




Original document

DATA CLUSTERING METHOD, AND APPLICATION THEREOF

Patent number: JP2002109536
 Publication date: 2002-04-12
 Inventor: RAHMAN SABBIR AHMED
 Applicant: NIGHTINGALE TECHNOLOGIES LTD
 Classification:
 - international: G06T7/00; G06F17/30
 - european:
 Application number: JP20000316799 20001017
 Priority number(s): EP20000308303 20000922

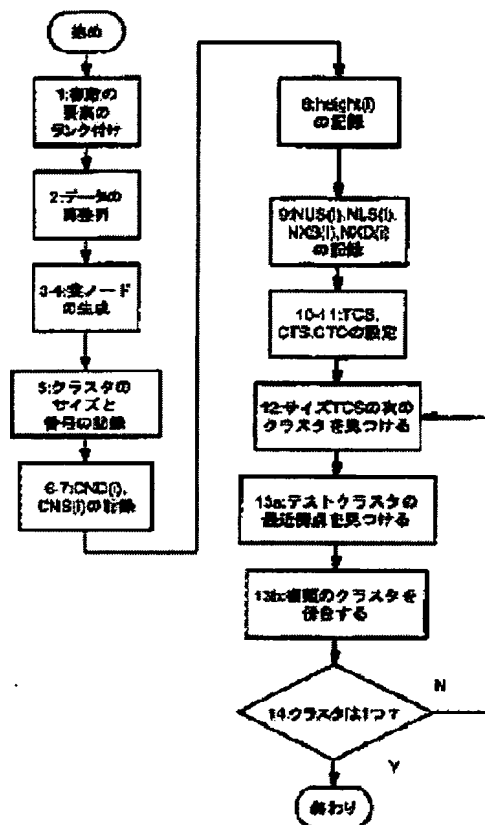
Also published as:

 EP1191459 (A1)
 WO0225574 (A3)
 WO0225574 (A2)

[View INPADOC patent family](#)[Report a data error here](#)

Abstract of JP2002109536

PROBLEM TO BE SOLVED: To provide a technology of improving the speed of generating clustering data for expressing the hierarchical clustering of a series of data samples. **SOLUTION:** In this technology, the size is increased to select other closest cluster, the data samples based on the absolute distance from the reference are merged and arranged, the data sample which is closest in the limited index range is searched, and a plurality of clusters are selected when comparing the distance by totaling the contribution from a plurality of components of each element in the order of areas between quartiles of a plurality of components of the data samples of each element.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-109536

(P2002-109536A)

(43) 公開日 平成14年4月12日 (2002. 4. 12)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード* (参考)
G 0 6 T 7/00	2 5 0	G 0 6 T 7/00	2 5 0 5 B 0 7 5
// G 0 6 F 17/30	2 1 0	G 0 6 F 17/30	2 1 0 D 5 L 0 9 6

審査請求 有 請求項の数36 O L 外国語出願 (全 65 頁)

(21) 出願番号 特願2000-316799(P2000-316799)

(22) 出願日 平成12年10月17日 (2000. 10. 17)

(31) 優先権主張番号 0 0 3 0 8 3 0 3. 7

(32) 優先日 平成12年9月22日 (2000. 9. 22)

(33) 優先権主張国 欧州特許庁 (E P)

(71) 出願人 500482256

ナイチンゲール テクノロジーズ リミテッド

ジブラルタル, メイン ストリート, センター プラザ, ユニット 2 ビー

(72) 発明者 サビール アフメッド ラフマン

ジブラルタル, メイン ストリート, センター プラザ, ユニット 2 ビー, ナイチンゲール テクノロジーズ リミテッド内

(74) 代理人 100094318

弁理士 山田 行一 (外1名)

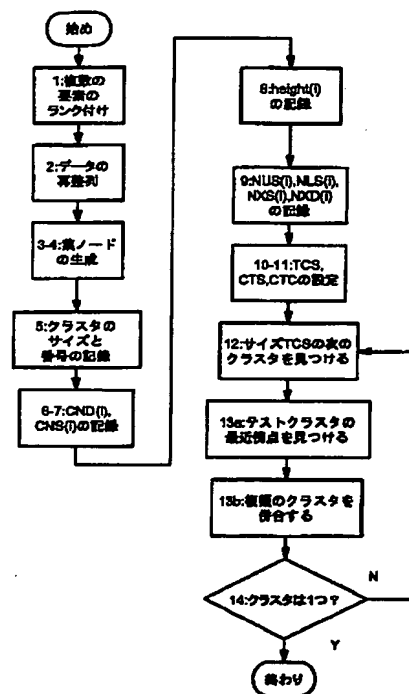
最終頁に続く

(54) 【発明の名称】 データクラスタリング方法とアプリケーション

(57) 【要約】

【課題】一連のデータサンプルの階層クラスタリングを表すクラスタリングデータを生成する速度を改善する技術に関する。

【解決手段】これらの技術には、他の最近傍のクラスタを選択するためにサイズを大きくし、基準からの絶対距離に基づくデータサンプルを併合して整列し、限定されたインデックス範囲内で最近傍のものを探索し、各要素のデータサンプルの複数の成分の4分位数間領域の順番に各要素の複数の成分からの寄与値を合計することによって距離の比較を行うときに行う複数のクラスタを選択する処理が含まれる。



【特許請求の範囲】

【請求項1】 階層データクラスタリング方法であって、

- a.一連のデータサンプルを入力値として受信する工程と、
- b.データサンプルの初期的クラスタ割当てを記録する工程と、
- c.データサンプルの各クラスタに対して、そのクラスタに最も類似するクラスタを決定し、それに対する相違度を所定の相違度関数に基づいて記録する工程と、
- d.相違度を決定したデータサンプルの個性値を記録する工程と、
- e.最も類似するクラスタと現在のクラスタを単一クラスタとして記録する工程と、
- f.所定の程度のクラスタリングが達成されるまで、工程cからeを繰返す工程と、
- g.前記記録された相違度とそれに関連するデータサンプルの個性値を出力として供給する工程とを備え、工程cでは、複数のクラスタがサイズの昇順に選ばれる方法。

【請求項2】 工程cでは、現在のクラスタ内のデータサンプルのどれに対しても最も類似するデータサンプルをもつクラスタを最も類似するクラスタとして確定する請求項1記載の方法。

【請求項3】 階層データクラスタリング方法であって、

- a.一連のデータサンプルを入力値として受信する工程と、
- b.所定の絶対距離の測定基準に基づいて、実質的に共通の基準からの絶対距離の順に、データサンプルセットにインデックスを付ける工程と、
- c.前記データサンプルの初期的クラスタ割当てを記録する工程と、
- d.データサンプルの各クラスタに対して、所定のサンプル間距離測定基準に基づいて現在のクラスタに対する最近傍のクラスタを決定して、それに対する距離を記録する工程と、
- e.前記距離を決定したデータサンプルの個性値を記録する工程と、
- f.最近傍クラスタと現在のクラスタを単一クラスタとして記録する工程と、
- g.所定の程度のクラスタリングが達成されるまで、工程dからfを繰返す工程と、
- h.前記記録された距離とデータサンプル個性値を出力値として供給する工程とを備え、工程dは、現在のクラスタ内の各データサンプルと距離の比較を行うために、現在のデータサンプルのものより大きなインデックス値をもつ上位のインデックスと現在のデータサンプルのものより小さなインデックス値をもつ下位のインデックスの間のインデックス範囲内の現在のクラスタの外にある

データサンプルのサブセットだけを選択することを含む方法。

【請求項4】 工程fは、もし最近傍クラスタと現在のクラスタのデータサンプルがインデックス上で近傍でないなら、最近傍クラスタと現在のクラスタのデータサンプルが近傍になるようにデータサンプルの幾つかにインデックスを付け直す処理を備える請求項3記載の方法。

【請求項5】 下位のインデックスのデータサンプルの絶対距離と現在のデータサンプルのものとの差と、上位のインデックスのデータサンプルの絶対距離と現在のデータサンプルのものとの差の小さい方が、現在のデータサンプルとインデックス範囲内だが現在のテストクラスタ内ではないデータサンプルとの間の最小サンプル間距離より大きくなるように、上位と下位のインデックスを決定する請求項3又は請求項4の何れか1項記載の方法。

【請求項6】 現在のデータサンプルとインデックス範囲内だが現在のテストクラスタ内ではないデータサンプル間の最小サンプル間距離より前記差の小さい方が大きくなるまで、下位のインデックスのデータサンプルの絶対距離と現在のデータサンプルのものとの差が上位のインデックスのデータサンプルの絶対距離と現在のデータサンプルのものとの差より小さいか、もしくは、大きいかに基づいて連続的に下位のインデックスを減らすか、もしくは、上位のインデックスを増やすことによって、上位と下位のインデックスを決定する請求項5記載の方法。

【請求項7】 前記絶対距離測定基準は最大の変化量をもつ複数のデータサンプルの要素の成分差である請求項3～6の何れか1項記載の方法。

【請求項8】 前記変化量は、データサンプルの所定の一部が入る範囲として決定される請求項7記載の方法。

【請求項9】 階層データクラスタリング方法であって、

- a.各々が複数の要素をもつ一連のデータサンプルを入力値として受信する工程と、
- b.前記複数の要素の各々に対して、前記データサンプルの要素の変化量の測定値を決定する工程と、
- c.前記データサンプルの要素をそれらの変化量の測定値に基づいて整列する工程と、
- d.各データサンプルをそれ自体のクラスタに属するものとして、初期設定する工程と、
- e.各クラスタを順に選び、そのクラスタのサンプルの最近傍だがそのクラスタの一部をまだ構成しないデータサンプルを決定する工程と、
- f.最近傍のデータサンプルのクラスタと現在のクラスタを併合する工程と、
- g.所望の程度のクラスタリングが達成されるまで工程e)とf)を繰返す工程とを備え、前記変化量の測定値は、その要素の最大値と最小値を除いた複数のデータサ

ンプルの所定の一部の範囲にある方法。

【請求項10】 階層データクラスタリング方法であって、

- a.一連のデータサンプルを入力値として受信する工程と、
- b.前記データサンプルの初期的クラスタ割当てを記録する工程と、
- c.データサンプルの各クラスタに対して、現在のクラスタに最も類似するクラスタを決定して、データサンプルの複数の要素の所定の相違度関数に基づいてその相違度を記録する工程と、
- d.相違度を決定したデータサンプルの個性値を記録する工程と、
- e.前記最も類似するクラスタと前記現在のクラスタを単一クラスタとして記録する工程と、
- f.所定の程度のクラスタリングが達成されるまで工程cから工程eを繰返す工程と、
- g.前記記録された相違度とそれに関連するデータサンプルの個性値を出力値として供給する工程とを備え、工程cは、各相違度演算に対し、各要素内のデータサンプルの変化量の降下順に各要素の距離測定値の成分を選び、累積的相違度値を計算し、もし累積した相違度値が相対的相違度値を越えるならば前記相違度の計算を終了をさせる処理を備える方法。

【請求項11】 a.上述の請求項に基づく方法を実行し、

- b.データサンプルのセットと工程aの方法の出力値に基づき圧縮されたデータセットを生成し、
- c.前記圧縮されたデータを出力するデータ圧縮方法。

【請求項12】 前記圧縮されたデータセットを格納することを備える請求項11記載の方法。

【請求項13】 前記圧縮されたデータセットを転送することを備える請求項11記載の方法。

【請求項14】 a.請求項1～10の何れかの方法を実行し、

- b.工程aの方法の出力値に基づいて、同じクラスタ内の前記複数のデータサンプルのものの間の関連を示すことを備える特徴抽出方法。

【請求項15】 工程bは、少なくとも1つのデータサンプルの複数の特性と少なくとも1つのデータサンプルのクラスタリング特性に基づく所定の分類データを比較し、前記比較の基礎をなす同じクラスタ内の複数のデータサンプルの分類値を出力することを備える請求項14記載の方法。

【請求項16】 a.請求項1～10の何れか1つの方法を実行し、

- b.工程aで決定されたデータサンプルのクラスタリング特性に基づいて前記複数のデータサンプルのものの少なくとも2つの特徴的特性を決定し、
- c.前記複数のデータサンプルの少なくとも1つのに対

して、前記特徴的特性の混在割合を決定する、非混在方法。

【請求項17】 前記特徴的特性のうちの1つをもつ前記複数のデータサンプルのものの間の少なくとも1つの境界領域を決定することを備え、工程cでは、少なくとも1つのデータサンプルが境界領域内に存在する、請求項16記載の方法。

【請求項18】 前記境界領域は、空間的、もしくは、時間的なエッジ検出によって決定される請求項16記載の方法。

【請求項19】 前記複数の特徴的特性の混在割合を構成しないものとして決定された値をもつ境界領域内の複数のデータサンプルを例外として示すことを備える請求項17又は18の何れか1項記載の方法。

【請求項20】 a.請求項1～10の何れかの方法を実行し、

- b.以降の処理のために前記複数のデータサンプルのサブセットを、工程aで決定されたそれらのクラスタリング特性に基づいて選択することを備えるデータ選択方法。

【請求項21】 前記サブセットを選択することで、前記クラスタ内の所定のクラスタリング評価基準に基づくクラスタを備える請求項20記載の方法。

【請求項22】 前記サブセットを選択することで、その他のクラスタに関連する所定のクラスタリング評価基準に基づくクラスタを備える請求項20記載の方法。

【請求項23】 前記複数のデータサンプルの少なくとも1つを前選択することを備え、前記サブセットを選択することで、前記、もしくは、各前選択されたデータサンプルを含むクラスタを備える請求項20記載の方法。

【請求項24】 前記複数のデータサンプルの少なくとも1つを前選択することを備え、前記サブセットを選択することで、前記、もしくは、各前選択されたデータサンプルを除いたクラスタを備える請求項20記載の方法。

【請求項25】 a.請求項1～10までの何れか1つにて請求された方法を実行する工程であって、前記複数のデータサンプルはネットワークの複数のノードを表す、当該工程と、

- b.工程aの出力値によって定義される最小全長木に基づき、前記ネットワークの前記複数のノード間の接続表現を生成する工程とを備える、ネットワークデザインを生成する方法。

【請求項26】 a.請求項1～10までの何れか1つの方法を実行する工程であって、前記複数のデータサンプルはネットワークの複数のノードを表す、当該工程と、

- b.工程aの出力値によって定義される最小全長木に基づく前記複数のノード間の接続を生成する工程とを備えるネットワーク構成方法。

【請求項27】 a. テストサンプルに最も類似するクラスタのデータサンプルを決定する工程と、
b. 前記と前記最も類似するデータサンプルの相違度と、前記最も類似するデータサンプルと前記クラスタ内のその他のデータサンプルの相違度に基づいて、前記テストサンプルと前記クラスタに関連する値を計算する工程と、
c. 前記クラスタに対する計算値に基づいて次の処理を実行する工程とを備え、

工程bでは、前記テストサンプルと前記最も類似するデータサンプルの相違度と前記テストサンプルと前記クラスタ内の前記最も類似するデータサンプルに最も類似する他のデータサンプル間の相違度の関数として前記値を計算する、複数のデータサンプルを含むクラスタにテストサンプルを分類する方法。

【請求項28】 少なくとも3つのデータサンプルを含むクラスタにテストサンプルを分類する方法であって、
a. 前記クラスタ内の複数のデータサンプルの複数のペア間の相違度に基づいて、前記テストサンプルと前記クラスタに関連する値を計算する工程と、

b. 前記クラスタに対して計算された値に基づいて次の処理工程を実行する工程を備え、工程aでは、最も類似するデータサンプルに接続されたエッジよりも小さい、最も大きな相違度をもつ最小全長木内のエッジの相違度よりも前記テストサンプル相違度が小さくなければ、前記テストサンプルと前記クラスタ内で最も類似するデータサンプルとのテストサンプル相違度の関数として前記値を計算する方法。

【請求項29】 さらに1つ以上のクラスタの各々に関するテストサンプルに関連づけられた値を計算することを備え、

前記次に処理工程は前記クラスタの各々に対して計算された複数の値間を比較することに基づく請求項27又は請求項28の何れか1項記載の方法。

【請求項30】 a. テストサンプルを受信し、
b. 請求項27～29の何れか1つに基づく方法を実行するボタン認識方法。

【請求項31】 a. テストサンプルを受信し、
b. 請求項1～10の何れか1項に基づく方法を実行し、

c. 請求項27～29の何れか1項に基づく方法を実行するボタン認識方法。

【請求項32】 前記複数のデータサンプルは、物理的特性のサンプルである請求項1～31の何れか1項記載の方法。

【請求項33】 適切に構成されたプロセッサによって実行されるときに、請求項1～32の何れか1項に基づく方法を実行するように構成されたコンピュータプログラム。

【請求項34】 請求項33のコンピュータプログラム

を実行するキャリア。

【請求項35】 請求項1～32の何れか1項に基づく方法を実行するように構成された装置。

【請求項36】 データ入力部、前処理部、請求項1～10の何れか1項に基づく方法を実行するように構成されたクラスタプロセッサ、後処理部、データ出力部を備える装置。

【発明の詳細な説明】

【0001】本発明の分野本発明の1つの態様は、階層データクラスタリング方法と、この方法を含む処理とアプリケーションに関する。本発明の他の態様は、1つ以上のクラスタにテストサンプルを分類する方法に関し、さらに、この方法に関する処理とアプリケーションに関する。

【0002】本発明の背景

階層クラスタ分析は、所定の分類を行うことなく複数のサンプル間の類似度に基づくクラスタ構造に一連のデータサンプルを分類することに関する。クラスタ分析の方法とアプリケーションについて多くの文献がある。その例としては、

'データクラスタリング: リビュー', ジェーン a. K., マーティ, M.N., フィン, P.J. 著, ACM コンピューティングサーベイ, No.3, 31巻, 264頁

'分類', ゴードン, a.D., 第3章, チャプマン&ホール出版, 1981年

単一リンククラスタ分析はクラスタ分析の一種であり、一連のデータサンプルの'最小全長木'を探索する処理が含まれる。'最小全長木'は、複数のエッジの全長、もしくは、'重さ'が最小になるようにデータサンプルの複数のペアを合体させる一連のライン、即ち、'複数のエッジ'である。例えば、以下の文献を参照されたい。

【0003】'アルゴリズム入門(概論)', コーメン, T.E., レイザーソン, C.E., ライバート, R.L. 著, 第24章, MITプレス出版, 1990年

'アルゴリズム', セジウィック, R. 著, 第31章, 第2版, 1988年, アディソンウエズレイ出版

実用的アプリケーションでのクラスタ分析技術の有効性は速度と格納条件に関する効率に依存するが、これはデータサンプル数N、各サンプルのD次元の数、データサンプル構成の関数である。最悪の場合は、階層クラスタリングアルゴリズムは N^2 のオーダーの時間を必要とするため、大きなデータセットに対しては実用的ではない。

【0004】本発明者の著書である'ハイバースペクトル画像処理のための効率的/対話的/集積的階層クラスタリングアルゴリズム', 撮影分光測定法SPIE会議プロシーディングからの出版, カリフォルニア州サンディエゴ, 1998年7月, SPIE3438巻, 210-221頁, にはクラスタリングアルゴリズムに関する記述があり、このアルゴリズムでは、近傍のポイントが近

傍のインデックスを備えるようにデータポイントにインデックスを付け、限定されたサブ空間を探索して複数のポイントペア間で必要な比較処理の数を削減する。しかしながら、1方向だけの探索を行うため、単一リンククラスタリングを正確に行うことができない。

【0005】欧州特許公開EP913780aではデータクラスタリング方法について開示しており、この方法では、最近傍のサンプルを選択する前に、注目のサンプルの最近傍でありそうもないデータサンプルを除去することによって距離演算の総数を削減する。

【0006】本発明の発明者の著書である'生物医学的撮影に適用可能な分散されたエンドメンバのためのスペクトル非混合アルゴリズム'、SPIEプロシーディング、第3438巻では、データポイントを階層クラスタリングすることによって、一連のデータポイントに属するテストポイントの尤度値を演算する方法について開示している。

【0007】本発明の概要

本発明の一態様によれば、以下の工程を備えるデータクラスタリング方法が提供される。

【0008】a.一連のデータサンプルを入力値として受信する。

【0009】b.各データサンプルをそれ自体のクラスタに属するものとして初期設定する。

【0010】c.各クラスタを順に選び、そのクラスタのサンプルに最も近いが、まだそのクラスタの一部を構成しないデータサンプルを決定する。

【0011】d.最近傍のデータサンプルのクラスタと現在のクラスタを合体する。

【0012】e. 所望の程度までクラスタリングが達成されるまで、工程c、dを繰返す。ここで、工程cでは、クラスタのサイズが大きくなる順にクラスタが選ばれる。この方法の利点は、距離数、即ち、演算に必要な相違度の測定をかなり減らせることである。何故ならば、一般的に、現在の最小クラスタより両方ともに大きい複数のクラスタのサンプル間の距離よりも、現在の最小クラスタのサンプルとそれより大きな他のクラスタのサンプル間の距離が小さいからである。

【0013】本発明の他の態様によれば、以下の工程を備えるデータクラスタリング方法が提供される。

【0014】a.一連のデータサンプルを入力値として受信する。

【0015】b.各データサンプルをそれ自体のクラスタに属するものとして初期設定する。

【0016】c.各クラスタを順に選び、そのクラスタ内のサンプルに最も近いが、まだそのクラスタの一部を構成しないデータサンプルを決定する。

【0017】d.最近傍のデータサンプルのクラスタを現在のクラスタに合体させる。

【0018】e. 所望の程度までクラスタリングが達成

されるまで工程c、dを繰返す。ここで、工程cでは、基準からの距離に基づいてインデックス付けられたデータサンプルのインデックス範囲によって定義された限定サブ空間を探索することによって、最近傍のデータサンプルを確定する。

【0019】この方法の利点は、基準からの距離に基づいてデータサンプルにインデックス付けすることによって最近傍のサンプルを含む注目のサンプルに対して最大/最小インデックスが付けられ、これにより、クラスタリング精度に妥協することなく注目のサンプルとの比較が必要なサンプルの数をかなり減らせることである。

【0020】本発明の他の態様によれば、以下の工程を備えるデータクラスタリング方法が提供される。

【0021】a.一連のデータサンプルを入力値として受信する。ここで、その各々は複数の要素をもつ。

【0022】b.前記要素の各々に対して、その要素のデータサンプルの差の測定値を決定する。

【0023】c.それらの差の測定値に基づいて複数のデータサンプルの要素を格納する。

【0024】d.各データサンプルをそれ自体のクラスタに属するものとして初期設定する。

【0025】e.各クラスタを順に選び、そのクラスタ内のサンプルに最も近いが、まだそのクラスタの一部を構成しないデータサンプルを決定する。

【0026】f.最近傍のデータサンプルのクラスタを現在のクラスタに合体させる。

【0027】g. 所望の程度までのクラスタリングが達成されるまで工程e)、f)を繰返す。ここで、差の測定値は、その要素の最大/最小値を除く複数のデータサンプルの所定の部分の範囲を示す。

【0028】この方法の利点は、複数のサンプル間の相違度を確定するときに最も重要になりそうな要素について最初に検討するため、複数のサンプルを比較するときに全要素について検討する必要がしばしばなくなることである。

【0029】本発明の他の様態によれば、複数のデータサンプルを含むクラスタにテストサンプルを分類する方法を提供する。本方法は以下の工程を備える。

【0030】a.クラスタ内で、テストサンプルに最も類似するデータサンプルを決定する。

【0031】b.クラスタ内でテストサンプルと最も類似するデータサンプルの相違度と、その最も類似するデータサンプルとその他のデータサンプルとの相違度に基づいてテストサンプルとクラスタに関連する値を計算する。

【0032】c.そのクラスタに関して計算した値に基づいてさらに複数の処理工程を実行する。ここで、工程bでは、クラスタ内のテストサンプルと最も類似するデータサンプルの相違度と、その最も類似するデータサンプルに最も類似するその他のデータサンプルとテストサ

ンプルの相違度の関数としてその値が計算される。

【0033】この方法の利点は、その計算値が個々のサンプルではなくエッジを参照して計算され、エッジによって結合される複数のサンプル間の領域の値の差を滑らかにすることである。

【0034】本発明の他の態様によれば、少なくとも3つのデータサンプルを含むクラスタにテストサンプルを分類する方法を提供する。本工程は以下の工程を備える。

【0035】a. クラスタ内の複数のデータサンプルペア間の相違度に基づいて、テストサンプルとクラスタに関連する値を計算する。

【0036】b. クラスタに関して計算された値に基づいてさらに複数の処理工程を実行する。ここで、工程aでは、最も類似するデータサンプルに接続するエッジより小さな最大相違度をもつ最小全長木のエッジの相違度よりもテストサンプルの相違度が小さくない場合は、クラスタ内のテストサンプルと最も類似するデータサンプルとのテストサンプル相違度の関数としてその値を計算する。

【0037】この方法の利点は、次の最短エッジに近いためにクラスタの最も密な領域に近くない場合よりも、最短エッジに近い場合により大きな重みをテストサンプルに与えることである。

【0038】ここで、添付図面を参照して本発明の特定の複数の実施形態を説明する。

【0039】特定の実施形態の説明

本発明の一実施形態に係る方法が以下で説明される。それぞれがD個の要素を備えるN個のサンプルの配列が入力される。

【0040】工程1 - ランク要素

4分位数間領域、即ち、複数のサンプルの中間の50%を含む第1の4分位数と第3の4分位数間の領域が、D個の要素の各々について計算される。この場合、4分位数間領域は有利な測定基準である。何故ならば、その領域の末端にあるまばらなサンプルに影響されないため、大多数のサンプルの違いをうまく表現できるからである。サンプル配列の複数の要素は4分位数間領域の降順に再整列されるか、もしくは、複数の要素のランクづけの順序が1つの要素ランク配列に格納される。

【0041】工程2 - データサンプルの再整列

以下の2つの方法のうちの一方を使って複数のデータサンプルを再整列させる。

【0042】2a) 半径方向の整列: 原点からの距離が増大する順に複数のサンプルを再整列させる。尚、例えば、最も大きな4分位数間領域の要素の最小成分をもつサンプルとなる原点を選択する。

【0043】2b) 線形整列: 選ばれた要素(最も大きな4分位数間領域をもつ要素が望ましい)の値が増大する順に複数のサンプルを再整列させる。何れの場合で

も、原点のインデックス値を1つの要素配列内に格納することで個々のサンプルを識別することができる。

【0044】工程3 - 二進木の葉ノードの生成
各サンプルが対応する二進木の葉ノードに割り当てられるので、再整列後のi番目のサンプルが二進木のi番目の葉ノードに初期的に割り当てられ、その割当て値は配列に格納される。

【0045】工程4 - クラスタラベルの生成
クラスタラベル配列を生成して、各サンプルが属するクラスタを指示する。各サンプルは初期的にはそれ自体のクラスタに属すると考えられるため、初期的にはクラスタラベルはサンプルのインデックス番号である。

【0046】工程5 - クラスタの大きさとその数の記録
各クラスタの大きさ(サンプルの数)とクラスタ数(初期的にはN個)が記録される。

【0047】工程6 - 最近傍の距離の記録
別のクラスタ内の各非併合サンプルから最近傍サンプルへの距離を変数CND(i)(現在の最近傍距離)として格納する。

【0048】工程7 - 最近傍のサンプルの記録
最近傍のサンプルのインデックスを整数値CNS(i)(現在の最近傍サンプル)として格納する。

【0049】工程8 - 併合の高さの記録
各非併合サンプルからそれが併合されるクラスタまでの距離が'併合の高さ'として格納される。初期的には併合処理が行われないため、距離は無限大(即ち、最大値)として設定される。

【0050】工程9 - サンプル間距離と次の距離を記録

各サンプルに対して、同じクラスタ内ではない次の最高位/最低位の葉ノードインデックスをもつ複数のサンプルが探索される。これらはそれぞれ、'次の上位'、'次の下位'のサンプルNUS(i)、NLS(i)と呼ばれる。例えば、もしテストサンプルが葉ノードiであれば、初期的には、次の上位のサンプルNUS(i)は葉ノードi+1にあり、次の低位のサンプルNLS(i)は葉ノードi-1にある。何故ならば、各サンプルは初期的にはそれ自体のクラスタに属するからである。

【0051】各サンプルとその次の上位のサンプルと下位のサンプルに対して、原点からの'絶対距離'が計算される。もし半径方向の整列を工程2a)で行うと、'絶対距離'は選ばれた原点からの半径方向の距離である。

もし線形整列が工程2b)で行われたならば、'絶対距離'は、選ばれた原点から選択された方向への距離である。尚、この原点は選択された方向で最小の成分をもつサンプルであることが望ましい。

【0052】次に、サンプル絶対距離とその次の上位のサンプルNUS(i)の絶対距離の差と、サンプルの絶対距離とその次の低位のサンプルNLS(i)の絶対距

離の差が計算される。これらの2つの差の小さい方がサンプルの'次の距離'NXD(i)として格納され、また、その2つの差の小さい方を与えた、次の上位のサンプルNUS(i)、もしくは、次の低位のサンプルNLS(i)のサンプルインデックスが、'次のサンプル'NXS(i)として格納される。

【0053】工程10 - テストクラスタサイズの設定
テストクラスタサイズTCSを初期的に1に設定する。

【0054】工程11 - 現在のテストサンプルとクラスタの設定

現在のテストサンプルCTSを初期的にサンプルインデックス1に設定し、現在のテストクラスタCTCはそのサンプルを含むクラスタ(初期的にはクラスタ1)である。

【0055】工程12 - テストクラスタサイズの次のクラスタを見つける

二進木で現われる順番に複数のクラスタを調べて、テストクラスタサイズTCSに等しいサイズの次のクラスタを見つける。この処理は以下のように実行される。

【0056】12a) もし現在のクラスタサイズがテストクラスタサイズに等しいなら工程12は終了する。

【0057】12b) そうでなければ、木で連続する複数の葉ノード内に常にグループ化される現在のテストクラスタのすぐ次の葉ノードのサンプルにジャンプする。本サンプルを含むクラスタを現在のテストクラスタとし、工程12a)へ進む。最後の葉ノードに達した場合は、テストクラスタサイズを現在で最も小さいクラスタのサイズとして以下のように設定する。

【0058】12ba) 現在のテストクラスタサイズをインクリメントし、'現在の最小のテストクラスタサイズ'をN(即ち、可能な最大クラスタサイズ)に設定する。

【0059】12bb) 現在のテストサンプルを第1の葉ノードのサンプルとして設定する。

【0060】12bc) 現在のテストクラスタのサイズがクラスタサイズと同じ場合は工程13に進む。

【0061】12bd) もし現在のテストクラスタサイズが現在で最小のテストクラスタサイズより小さい場合は、その最小のテストクラスタサイズを更新し、そのテストクラスタを最小のサイズテストクラスタとして格納する。

【0062】12be) 現在のテストクラスタの後に木のサンプルがなければ、工程12bf)に進む。そうでなければ、現在のテストクラスタのすぐ次の木のサンプルにジャンプし、これを現在のテストサンプルとして工程12bc)に進む。

【0063】12bf) 現在のテストクラスタを最小サイズのテストクラスタにする。

【0064】工程13 - クラスタに含まれない最近傍のサンプルと併合

13a) 現在のテストクラスタの最近傍のサンプルだが、それ自体はテストクラスタに含まれないサンプルを以下のように見つける。

【0065】13aa) 最小距離MinDistを無限(即ち、最大値)として設定する。

【0066】13ab) 現在のテストクラスタの各サンプルに対して、以下の処理を行う。

【0067】13aba) もし現在の最近傍サンプルCNS(i)が現在のテストクラスタCTC(i)の要素でなく、また、現在の最近傍距離CND(i)が最小距離MinDistより小さいなら、最小距離MinDistを現在の最近傍距離CND(i)として設定し、現在のサンプルインデックスを'先頭部'とし、現在の最近傍のサンプルCNS(i)を'末端部'として格納する。

【0068】そうでなければ、次の距離NXD(i)が最小距離MinDistより小さい場合は、現在の最近傍の距離CND(i)をリセットして無限大値に設定して次の処理を行うことによって、現在の最近傍のサンプルCNS(i)と現在の最近傍の距離CND(i)を更新する。

【0069】13abaa) 工程9での現在のサンプルの次のサンプルNXS(i)と次の距離NXD(i)を求める。

【0070】13abab) もし次の距離NXD(i)が最小距離より小さくないなら、クラスタの次のサンプルを現在のサンプルとして工程13ab)に戻る。さもなければ、現在のサンプルから次のサンプルNXS(i)までの距離MeasDistを測る。もしMeasDistが最近傍の距離CND(i)より大きいなら、工程13abadに進む)。工程1で決定された順番で各要素の距離成分を測り、その成分値を2乗し、その2乗値を複数の2乗値の合計値に加算することによって比較を行う。各合計計算処理のあとで、その合計値を最近傍距離CND(i)の2乗値と比較し、もし大きければ、これ以上の項の合計を計算することなく処理は工程13abad)に進む。比較を行う本方法では不要な演算を避け、特にDが大きい場合の速度が改善される。

【0071】13abac) もしMeasDistが最近傍距離CND(i)より小さければ、最近傍サンプルCNS(i)と最近傍の距離CND(i)を更新して、それぞれ、次のサンプルNXS(i)とMeasDistにする。さらに、もしMeasDistがMinDistより小さいならば、MinDistを更新してMeasDistとし、現在のサンプルを'先頭部'として格納し、また、次のサンプルNXS(i)を'末尾部'として格納する。

【0072】13abad) 次のサンプルNXS(i)が次の上位の、もしくは、次の低位のサンプルであったかどうか依存して、次の上位の、もしくは、次の低位のサンプルNUS(i)、NLS(i)を更新する。もし

し次のサンプルNXS(i)が次の上位サンプルNUS(i)であったならば、次に上位のサンプルNUS(i)を更新して、テストクラスタでない現在の次の上位サンプルNUS(i)の後の次の上位葉ノードインデックスをもつサンプルとする。もし次のサンプルNXS(i)が次の低位サンプルNLS(i)であったならば、次の低位サンプルNLS(i)を更新して、現在のテストクラスタでない現在の次の低位サンプルNLS(i)の前の次の低位葉ノードインデックスをもつサンプルとする。新たな次の上位/下位のサンプルNUS(i)/NLS(i)考慮して、次のサンプルNXS(i)と次の距離NXD(i)を再計算し、13aba b)に進む。

【0073】現在のテストクラスタの最後のサンプルが処理されると、工程13bへ進む。13b)本段階では、'先頭部'と'末尾部'は最小全長木に結合されるサンプルであり、先頭部を含む現在のテストクラスタは末尾部を含むクラスタと併合されて、併合の高さが最小距離に設定される。より大きな値のラベルをもつクラスタがより小さな値のラベルに加えられる。この処理は以下の

【0074】13ba)二進木のサンプルの葉の位置を再整理させる。即ち、より小さなクラスタとより大きなクラスタを隣接させるように、より小さなクラスタと、より小さいクラスタとより大きなクラスタ間のサンプルが葉ノードの位置で交換される。この交換を反映させるために、各葉ノードに対して格納されるサンプルインデックスが更新される。より下位のクラスタラベルをより上位のクラスタのサンプルに割り当てることによって、より大きな値ラベルをもつクラスタがより小さな値のラ

ベルをもつクラスタに加えられる。
【0075】13bb)先頭部のサンプルインデックス、末尾部のサンプルインデックス、最小距離Mindist(併合の高さに等しい)、下位のクラスタのラベルがそれぞれsource(i)、dest(i)、height(i)、join(i)として配列に格納される。*

サンプル インデックス	1	2	3	4	5	6	7	8	9
CND(i):	2.24	2	2	3.61	4.12	1	1	1	2.24
CNS(i):	2	3	2	3	7	7	8	7	8
NXD(i):	2.24	2	2	3.61	4.47	1	1	1	2.24
NXS(i):	2	3	2	3	6	7	8	7	8

【0085】工程12では、サンプル1が現在のテストサンプルとして選択され、また、テストクラスタラベルが1に設定される。工程13では、サンプル2が図3のエッジaで示されるサンプル1に結合され、クラスタ2がクラスタ1に併合される。以下の情報が記録される。

【0086】

source(2)=1; dest(2)=2; height(2)=2.24; join(2)=1
次のサンプル3が工程12で現在のテストサンプルとな

*ここで、iは上位のクラスタのラベルである。

【0076】13bc)より下位のクラスタのサイズを格納する配列要素を、より上位のクラスタのサイズによって増大させる。

【0077】13bd)クラスタ数を減少させる。

【0078】工程14-繰返し

もし1つのクラスタだけが残っているなら処理を終了する。そうでなければ工程12へ進む。

【0079】source(i)、dest(i)、height(i)の配列は、複数のサンプルの二進木と最小全長木を定義するためには十分なものである。Join(i)は冗長な情報を備えるが、後段の処理工程を不要にすることができる。

【0080】特定の例

ここで、要素の数Dは2であり、複数のデータサンプル、即ち、'複数のボタン'xの数は9である簡単な例を示す図3から図5を参照して、上の方法の一例を説明する。図3は、以下の複数のデータサンプル値を示す。

【0081】x(1)=(0,1); x(2)=(7,5); x(3)=(3,7); x(4)=(5,1); x(5)=(2,0); x(6)=(8,6); x(7)=(7,6); x(8)=(2,2); x(9)=(9,8)

工程1)では、4分位数間領域がx、yの両方に対して同じであるとき、複数の要素を再整理させる必要はない。

【0082】工程2)では、工程2a)の半径方向に整理させる方法が使われる。サンプル(0,1)が原点として選択され、複数のサンプルが(1,5,8,4,3,2,7,6,9)に再整理され、図3でイタリック体で示される新たな順序に再度インデックスが付けられる。もし工程2b)の線形整理方法を使って、x個の要素を選択したならば、配列は(1,5,8,3,4,2,7,6,9)となる。

【0083】工程6,7,9では、以下の値が得られる。

【0084】

【表1】

るとき、それは次のクラスタに属する。工程13では、サンプル3が図3のエッジbによって示されるサンプル2に結合され、クラスタ3がクラスタ1に併合される。以下の情報が記録される。

【0087】

source(3)=3; dest(3)=2; height(3)=2; join(3)=1

次のサンプル4は、工程12で現在のテストサンプルとなる。工程13では、サンプル4が図3のエッジcによ

って示されるサンプル3に結合され、クラスタ4がクラスタ1に併合される。以下の情報が記録される。

【0088】

source(4)=4; dest(4)=3; height(4)=3.61; join(4)=1
次のサンプル5は、工程12で現在のテストサンプルとなる。工程13では、サンプル5は図3のエッジdによって示されるサンプル7に結合され、クラスタ7はクラスタ5に併合される。これによって、工程13b)では、図3で示される木を現わすためにサンプル6とサンプル7間の葉ノードの位置の交換が必要となる。以下の

情報が記録される。

【0089】

source(7)=5; dest(7)=7; height(7)=4.12; join(7)=5
次のサンプル6は現在のテストサンプルとなる。何故ならば、それが次の葉ノードにあるからである。サンプル6が図3のエッジeによって示されるサンプル7に結合され、クラスタ6が(サンプル7が既に1つの要素である)クラスタ5に併合される。以下の情報が記録される。

【0090】

source(6)=6; dest(6)=7; height(6)=1; join(6)=5
次のサンプル8が現在のテストサンプルとなり、エッジfによって示されるサンプル7に結合される。クラスタ8がクラスタ5に併合され、以下の情報が記録される。

【0091】

source(8)=8; dest(8)=7; height(8)=1; join(8)=5
次のサンプル9が現在のテストサンプルとなり、エッジgで示されるサンプル8に結合される。クラスタ9がクラスタ5に併合され、以下の情報が記録される。

【0092】

source(9)=9; dest(9)=8; height(9)=2.24; join(9)=5
ここで、工程12では、テストクラスタサイズは4に上がるが、それは現在のクラスタの最小サイズである。また、クラスタ1が現在のテストクラスタとして選ばれる。サンプル1からサンプル4までが葉ノードの順に選ばれる。

【0093】テストサンプル1について、工程13a)では、最近傍サンプルCNS(1)が同じクラスタ内のサンプル2である。本方法は工程13a)に進む。次の下位サンプルがない場合は、次の上位サンプルNUS(i)と次のサンプルNXS(1)の両方がサンプル5である。次の距離NXD(i)= $\sqrt{45}$ であり、計測された距離MeasDistも同じ値である。これはMinDistと最近傍距離CND(1)より小さいので、CND(1)=MinDist=MeasDistであり、また、CNS(1)=5である。工程13abad)では、次の上位サンプルNUS(i)がサンプル6までインクリメントされ、本方法は工程13ababに戻る。次の距離NXD(1)はサンプル1とサンプル6の絶対距離間の差であり、 $\sqrt{65}$ である。これ

は、MinDistより小さくないので、次のテストサンプルのために工程13ab)に戻る。

【0094】テストサンプル2では、同じクラスタにある最近傍サンプルCNS(2)=3であり、本方法は工程13abaa)に進む。次の距離NXD(2)はサンプル2とサンプル5の絶対距離間の差であり、これは $\sqrt{45}-\sqrt{5}$ である。これは最小距離MinDistより小さいので、MeasDistはサンプル5までの距離として計算され、これは $\sqrt{50}$ である。これはCND(2)より小さいので、CNS(2)=5、CND(2)=MeasDistである。しかしながら、MeasDistはより小さくないので、本方法は工程13abad)に進む。NXS(2)は6になり、本方法は工程13ababへ戻る。NXD(2)は $\sqrt{65}-\sqrt{5}$ として計算される。これはMinDistより小さいので、MeasDistは $\sqrt{50}$ として計算される。これはCND(2)に等しく、MinDistより小さいので、本方法は工程13abad)に進む。NXS(2)は7になり、本方法は工程13ababに戻る。NXD(2)は $\sqrt{50}-\sqrt{5}$ である。これはMinDistより小さいので、MeasDistは $\sqrt{61}$ として計算される。これはMinDist、即ち、CND(2)より小さくないので、本方法は工程13abad)に進み、NXS(3)は8になり、本方法は工程13ababに戻る。NXD(2)は $\sqrt{89}-\sqrt{5}$ である。これはMinDistより小さくないので、本方法は次のテストサンプルのために工程13ab)に戻る。

【0095】テストサンプル3では、NXS(3)=5、NXD(3)= $\sqrt{45}-\sqrt{5}$ である。これはMinDistより小さいので、MeasDistは $\sqrt{26}$ として計算される。これはCND(3)とMinDistより小さいので、CNS(3)=5、MinDist=CND(3)=MeasDist= $\sqrt{26}$ である。工程13abad)では、NXS(3)が更新されて6となり、NXD(3)= $\sqrt{65}-\sqrt{5}$ である。これはMinDistより小さくないので、本方法は次のテストサンプルのために工程13ab)に戻る。

【0096】テストサンプル4では、CND(4)= $\sqrt{20}$ であり、これはMinDistより小さい。従って、MinDistはCND(4)となり、これがクラスタ内で最後のテストサンプルのとき本工程13a)は終了する。

【0097】工程13b)では、サンプル4は先頭部であり、サンプル6は末尾部である。対応するエッジは図3でhとして示される。クラスタ5はクラスタ1に併合され、以下の情報が記録される。

【0098】

source(5)=4; dest(5)=6; height(5)=4.47; join(5)=1
1つのクラスタだけが残されるので、本クラスタリング方法は停止する。本クラスタリング方法による出力に

は、以下の配列 (source(i)、dest(i)、height(i)、join(i)) が含まれる。

* 【表2】

Index <i>i</i>	source(<i>i</i>)	dest(<i>i</i>)	height(<i>i</i>)	join(<i>i</i>)
2	1	2	2.24	1
3	3	2	2	1
4	4	3	3.61	1
5	4	6	4.47	1
6	6	7	1	5
7	5	7	4.12	5
8	8	7	1	5
9	9	8	2.24	5

【0100】技術的处理

上述の方法は、クラスタリングアルゴリズム、好適には単一リンクアルゴリズムに係る処理、もしくは、アプリケーションに適用可能である。しかしながら、上の方法ではほとんど処理を必要としないので、周知の単一リンククラスタリング方法よりも非常に高速に所定のプラットフォームで実行可能である。本方法を物理量サンプルである物理データサンプルに適用することができる。従って、その方法からもたらされた出力は、基本的物理量の物理構成を表す。

【0101】周知のアプリケーションの幾つかが以下で説明され、また、それらは一般的な処理の種類に分類される。図6は装置の一般的な構成を示す。これらの処理を実行するために、データ入力部I、前処理部PRE、クラスタリングプロセッサCP、後処理部POST、データ出力部Oを備える。これらの処理部では離散的な物理成分を表す必要はない。データ入力部Iは、センサ、もしくは、センサアレイであり、非物理データの場合は、データ入力デバイス、もしくは、このデバイスのネットワークである。後段の処理工程の前に、入力データを記憶手段に格納することができる。前処理部PREは、必要ならばアナログ-デジタル変換を行うことができ、また、複数のデータ要素をクラスタリングに必要なものに限定することができる。クラスタリングプロセッサCPは1つ以上の物理プロセッサでもよく、本クラスタリング方法を実行し、クラスタリングデータを出力する。次の処理のために、後処理部POSTはクラスタリング構成に調和する複数のクラスタに単一階層クラスタを分割することができ、また、クラスタリング構成やクラスタ内データサンプルの複数の特徴に基づいて複数のクラスタの自動分類を行うことができる。データ出力部Oは、ディスプレイ、プリンタ、データ記憶手段等であり。

【0102】適切に構成されたプロセッサ、例えば、クラスタリングプロセッサCPによって実行されるときに本発明に係る方法を実行するプログラムを本発明の実施形態が備える。プログラムを取り外し可能なディスクや固定ディスク、テープ、もしくは、その他の記憶手段等

のキャリアに格納したり、キャリアとの電磁気信号の送受信を行うことができる。

【0103】圧縮

一連のデータサンプルを複数のクラスタの階層木に分類すると、データサンプル自体を、複数のクラスタの構成を記述する圧縮されたデータセットによって置き換えることができる。損失的圧縮を行う場合は、圧縮されたデータセットは個々のデータサンプルを特定せずに、複数のクラスタの一般的構成を表す。例えば、個々が所定値以下の併合の高さをもつ複数のクラスタに木を分けることによって、各クラスタ内の複数のデータサンプルは圧縮データセットのクラスタの重心座標によって代表される。損失なしの圧縮の場合は、各々が所定値より小さい併合の高さをもつ複数のクラスタに木を分けることによって、クラスタ内の個々のサンプルをクラスタの重心からの微分ベクトルによって代表させる。微分ベクトルは複数サンプルの絶対座標より小さい範囲をもつため、それをわずかな数のビットを使って表現することができる。いかなる種類のデータに対しても、即ち、画像、音声、映像等の物理データであろうと、人間が直接知覚できない量であろうと、経済上のデータ等の非物理データであろうと本技術を適用することができる。圧縮されたデータセットを記憶媒体に格納することで効率的な記憶がなされ、また、通信リンクやローカルバスを介して転送することで帯域幅条件を緩和することができる。

【0104】従って、本処理を実行するために、データ出力部Oがデータ格納部、もしくは、データ経路である装置が提供される。

【0105】セグメンテーションと特徴抽出

一連のデータサンプルが複数のクラスタの階層木に分類されると、例えば、木が分割される併合の高さを設定することによって木を別々のクラスタに分けることができる。各クラスタは異なるクラスのデータを代表するものであり、それを使って各クラスタに異なる特性を属性づけることによって分析することができる。各データサンプルのクラスタへの帰属を示すことができる。これは、例えば、それらのクラスタに基づいて表示される複数のサンプルを色符号化することによってなされる。

【0106】リモートセンシングの分野では、同様の技術を使って、異なる対象、もしくは、地形の種類を表示することができる。この表示はユーザが解釈することが可能であり、また、例えば、周知の対象、もしくは、地形の種類、形状やスペクトル特性を比較することによって自動的に識別可能である。

【0107】同様の技術を画像のセグメンテーションに適用することができる。ここで、例えば、カラー画像をグレースケール画像に変換したり、ビットマップ画像をベクトル画像に変換したりするために、画像は類似する色や形状の領域に分けられる。また、グレースケール画像やベクトル画像は、元画像より少ないビット数ですむので、本アプリケーションも圧縮例の1つである。

【0108】幾つかのケースでは、データサンプルセットの複数のセグメントがオーバーラップするため、オーバーラップ領域に存在する種類の各セグメントの割合を推定する必要がある。例えば、リモートセンシング画像内の木や草等の画像対象の2つの主成分を混合した領域が画像内にある。密に集まったデータサンプルを探すことによって、1つの成分だけを含む純粋な領域を最初に識別し、純粋な成分のスペクトル特性を確定する。次に、エッジ検出を行って、これらの純粋な領域間の境界を識別する。これらの境界領域では、純粋な領域のスペクトル特性と境界領域のスペクトル特性の混在モデルのフィッティングを行って、各成分の割合を決定する。本技術の一例が、上で引用された論文'生物医学的撮影に適用可能な分散されたエンドメンバのためのスペクトル非混合アルゴリズム'で記述されている。

【0109】もし境界領域のデータサンプル特性が所定の許容値内で混在モデルと一致しないなら、データサンプルに対して例外としてのフラグを付け、ディスプレイ上でハイライトすることができる。例外検出は、腫瘍等の小さな異常部を検出するための医療撮影と、異常な対象、もしくは、特性を検出するためのリモートセンシングに役に立つ。

【0110】幾つかの特定のアプリケーションでは、複数のサンプルの空間的、もしくは、時間的特性によって複数の純粋なサンプル間の境界を決定する。しかしながら、要素空間での複数のサンプルのクラスタリング特性に純粋に関連して境界を定義することができる。そのため、その空間の密なクラスタには純粋なサンプルが含まれると考えられる。要素空間のクラスタ間のサンプルは混在サンプルと考えられる。

【0111】従って、本処理を実行する装置の後処理工程POSTでは、個々のクラスタに関連づけられたデータフラグ、もしくは、ラベルを生成することができる。また、データ出力部Oはデータフラグ、もしくは、偽色、もしくは、データサンプルの輪郭表示等によるラベルの示値を供給する。

【0112】データマイニング/ブラウジング

本技術では、クラスタ構成を使って、大きなデータ集合のサブセットを検査するために選択する。1つのケースでは、1つの初期的データサンプルが見つけれられ、初期的データサンプルが属するクラスタの他の要素が検査のために選択される。本技術の1つのアプリケーションは文書探索や検索、例えば、ウェブ探索の分野である。

【0113】その他のケースでは、密な集まり（例えば、低い併合の高さの多数のサンプル）等の所望の特性をもつ複数のクラスタが選択され、選択されたクラスタの要素が検査される。本技術の1つのアプリケーションは、データマイニングの分野である。この分野では、大規模データベースの密なクラスタを選択し分析して、クラスタの要素に基づく推論を行う。所望の特性は非常にゆるい集まりであってもよい。例えば、偽者を検出する分野では、他のサンプルと異なるデータサンプルが詐欺的活動を示すことがある。

【0114】データサンプルが非測定用（即ち、それらは測定値の整列構成をとらない）である場合は、2つのデータサンプルの差を数値表現するために相違度関数を選ぶ必要がある。相違度関数は、例えば、2つの文書に現われる類似する単語の数の関数でよい。

【0115】従って、本処理を実行する装置のデータ入力部Iはデータベースでよく、また、データ出力部Oは、選択されたデータサブセットを識別する、例えば、ターミナルの処理部でよい。

【0116】ネットワーク設計

総エッジ長、もしくは、総重量を最小化するように各データサンプルを少なくとも1つのその他のサンプルに接続したネットワークを最小全長木が表す。従って、この最小全長木は、ネットワーク内で複数のノードを最大効率で接続する必要がある現実問題に対する最適解である。この問題には回路設計が含まれる。ここでは、複数のデータサンプル間の距離が複数の回路ノードを接続するために必要な配線の長さを表す。同様に、複数のデータサンプルが複数の通信ノードを表し、また、それらを接続することによって得られるそれらの間の距離が不効率さを表す場合に、最小全長木は複数のノードを接続する最も効率的な方法を表す。

【0117】上の方法に基づいて最小全長木を探索する方法をこのような現実の問題に適用することができる。

【0118】従って、本処理を実行する装置のデータ入力部Iは、接続される複数のノードの特性を表すデータファイルを供給することができ、また、データ出力部Oは複数のノードを接続するための設計値を表すことができる。この設計値は、設計を行うためのグラフィック表現、もしくは、一連のインストラクションでよい。設計値に基づいて接続を生成するために、一連のインストラクションを自動的に実行することができる。

【0119】ボタン認識

ボタン認識には、既に分類された一連のデータサンプル

との類似度に基づいて新たなデータサンプルを分類する処理が含まれる。

【0120】クラスタランク関数

パターン認識は、データサンプルの所定のクラスタに対するランク関数を生成するために役に立つ。ランク関数はデータサンプルの要素の関数であり、新たなデータサンプルのランク値を所定のクラスタの要素として供給する。ランク値を使ってどのクラスタの新たなデータサンプルを分類すべきかを決定することができる。

【0121】ここで、最小全長木を定義するための上で 10
定義されたデータが与えられた場合での所定のクラスタ*

Index <i>i</i>	source(<i>i</i>)	dest(<i>i</i>)	height(<i>i</i>)	join(<i>i</i>)
2	6	7	1	5
3	8	7	1	5
4	3	2	2	1
5	1	2	2.24	1
6	9	8	2.24	5
7	4	3	3.61	1
8	5	7	4.12	5
9	4	6	4.47	1

【0124】図7で示されるようにこの結果を表すことができる。ここでは、併合線が交差しないように複数のサンプルが再整理される。

【0125】ランク関数の輪郭

説明を容易にするために、図8を参照してランク関数の輪郭をここで説明する。新たなデータサンプルに対するランク関数値を計算するために輪郭形状を計算する必要はない。

【0126】まず、最小のheight(*i*)に等しい半径の 30
超球（我々の例では複数の球）が最小の高さで結合された各サンプルの回りに描かれる。我々の例では、複数の超球がサンプル6、7、8の各々の回りに描かれる。複数の超球がオーバーラップする周囲には、確率 $(N - y(1) - 1) / (N - 1)$ が割り当てられている。ここで、 $y(1)$ はその高さで形成されたエッジの数である。この場合、確率は $6/8$ である。

【0127】次に、サンプル2と3を結合する次の最小 40
併合の高さ2へ進む。これらのサンプルの回りに、次に小さい併合の高さ(=1)に等しい半径の複数の超球が描かれ、その周囲に確率 $6/8$ がまた割り当てられる。次に、現在の併合の高さ以下のサンプルの全ての回りに現在の併合の高さの複数の超球を描き、それらの周囲に確率 $(N - y(2) - 1) / (N - 1)$ を割り当てる。ここで、 $y(2)$ は現在の併合の高さ以下で形成されたエッジの数である。この場合、確率は $5/8$ である。

【0128】次に、サンプル1、2、8、9にあてはまる 50
次の最小併合の高さ2.24に進む。それらの回りに、次に低い併合の高さ(=2)に等しい半径の複数の超球を描く。それらに対して、確率 $5/8$ がまた割り当

*に対するランク関数を生成する方法を説明する。最小全長木を定義するデータを上述のクラスタリング方法で得る必要はないが、それは処理速度の観点からは好ましいことである。

【0122】前処理 ー併合の高さの順に出力値を再整理する

次の処理を簡単にするために、出力配列が高さの順に再整理される。再整理処理の特定の例を以下に示す。

【0123】

【表3】

てられる。次に、現在の併合の高さ以下の全サンプルの回りに現在の併合の高さの複数の超球を描き、それらの周囲に確率 $(N - y(3) - 1) / (N - 1)$ を割り当てる。ここで、 $y(3)$ は現在の併合の高さ以下で形成されたエッジの数である。この場合、確率は $3/8$ である。

【0129】次に、サンプル3、4にあてはまる次の最小併合の高さ3.61に進む。それらの回りに、次に低い併合の高さ(=2.24)に等しい半径の複数の超球を描く。それらに対して確率 $3/8$ がまた割り当てられる。次に、現在の併合の高さ以下の全サンプルの回りに現在の併合の高さの複数の超球を描き、それらの周囲に確率 $(N - y(4) - 1) / (N - 1)$ を割り当てる。ここで、 $y(4)$ は現在の併合の高さ以下に形成されたエッジの数である。この場合、確率は $2/8$ である。

【0130】次に、サンプル5、7にあてはまる次の最小併合の高さ4.12に進む。それらの回りに、次に低い併合の高さ(=3.61)に等しい半径の複数の超球を描く。これらに対して、確率 $2/8$ がまた割り当てられる。次に、現在の併合の高さ以下の全サンプルの回りに現在の併合の高さの複数の超球を描き、それらの周囲に確率 $(N - y(5) - 1) / (N - 1)$ を割り当てる。ここで、 $y(5)$ は現在の併合の高さ以下で形成されたエッジの数である。この場合、確率は $1/8$ である。不十分な空間であるときは、これらの複数の円は図8では示されない。

【0131】最終的に、サンプル4、6にあてはまる最大併合の高さ4.47に到達する。それらの回りに、次に低い併合の高さ(=4.12)に等しい半径の複数の

超球を描く。それらに対して、確率 $1/8$ がまた割り当てられる。次に、現在の併合の高さ以下の全サンプルの回りに現在の併合の高さの複数の超球を描き、それらの周囲に確率 $(N-y(6)-1)/(N-1)$ を割り当てる。ここで、 $y(6)$ は現在の併合の高さ以下で形成されるエッジの数である。この場合、確率は $0/8$ である。不十分な空間であるときは、これらの複数の円は図8では示されない。

【0132】テストデータサンプルのためのランク関数値を計算するために、複数の円の周囲の間を補間する。もし中心が最小エッジの先頭部と末尾部であれば、最小半径の複数の円内の中心でランク関数=1までの補間を行う。さもなければ、次の最小の併合の高さの複数のサンプルに対して、ランク関数は最小半径の複数の円内で一定値となり、その境界の値に設定される。

【0133】ランクの推定-球の場合
ここで、超球の境界を定義した上述の第1の整列の場合での補間によるランク値の計算について以下で詳述する。

【0134】工程15 - 絶対距離の探索
各サンプルに対して、原点から半径方向の、もしくは、線形の'絶対距離'を上工程2と同様に計算する。

【0135】工程16 - 絶対距離の整列
サンプルの絶対距離セットを整列させ、インデックスを付ける。

【0136】工程17 - ランク値の計算
複数のデータサンプルが1つ以上のクラスタに分類される。例えば、特定の併合の高さで2進木を'切る'ことで、その併合の高さより上の全クラスタは様々なクラスタに属すると考えられる。もしくは、データサンプルを複数のグループに先験的に分けて、各グループ毎に別々にクラスタリングを行うことができる。

【0137】図11の概要で示されるように、各クラスタに対して、

17a) 現在のクラスタとテストサンプルに限定して1つのサンプルの最近傍のものを探索するために工程13a)に類似する方法を使って、テストサンプルの最近傍のクラスタからサンプルNSPを探索する。

【0138】17b) クラスタ内でテストサンプルから最近傍のサンプルNSPまでの距離 d と、クラスタ内の最大のエッジ長 e を求める。もしそのクラスタが1つのサンプルだけを含むためエッジがないなら、 $e=d/2$ とする。もし $d>e$ ならば、テストサンプルは完全にそのクラスタの外にあり、ランク値 $R=e-d$ と決定される。これは負となる。そして、本方法を停止する。

【0139】17c) テストサンプルはクラスタ内に存在する。もしクラスタに単一のサンプルだけが含まれるなら、 $R=1-d/e$ とし、停止する。

【0140】17d) クラスタは複数のサンプルをもつ。以下の複数の工程を実行する。

【0141】17da) Tをクラスタの最小全長木エッジの数とする。

【0142】17db) 複数のエッジにその長さの昇順にインデックスを付けることを考慮して、CEを現在のエッジのインデックスとする。CEは、 d と $(r-d)$ の大きい方に等しいか、それよりより大きい高さの第1のエッジに初期的に設定される。ここで、 r はNSPの併合の高さである。

【0143】17dc) NLEをエッジCEより長い第1のエッジのインデックスとする。

【0144】17dd) NHをエッジCEより短い長さのエッジの数とする。

【0145】17de) MINを、テストサンプルから現在までで考えられる最近傍のサンプルまでの距離とする。MINは最大値に初期設定される。

【0146】17df) SLを、エッジCEより短い、MST内で最長のエッジの長さとする。SLはゼロに初期設定される。

【0147】17dg) LLを現在のエッジCEの長さとする。LLはゼロに初期設定される。

【0148】17dh) NDを、長さがLLのクラスタの最小全長木内のエッジの数とする。NDは1に初期設定される。

【0149】17di) 現在のエッジCEに対するLLを求める。

【0150】17dj) NDを求める。現在、CEはLLより長い最短エッジのインデックスなので、CEをND分インクリメントする。

【0151】17dk) SL以下の併合の高さの全サンプルを初期的に含むものとして、'アクティブサンプル'の群を定義する。'アクティブ領域'ARは、アクティブサンプルASの距離SL内の全サンプル領域として定義される。

【0152】17dl) もしLLが最小全長木の最長のエッジ長以下なら、更新されたARを見つけ、テストサンプルがその中に入るかどうかを以下のように確認する。

【0153】17dla) 併合の高さLLのサンプルをアクティブサンプルASに加える。併合の高さLLの各サンプルに対して、テストサンプルとの距離TDを測る。もしTDがMINより小さいなら、MINをTDとSLの大きな方に設定する。

【0154】17dlb) もしMINがLL以下であれば、工程17dn)に進む。ここで、テストサンプルがアクティブな領域に存在するときのランク値を求める。

【0155】17dle) $SL=LL$ に設定し、LLを新たなエッジの長さとする。LLより短いエッジの数がND分増加したときにNDをNHに加える。NDを長さLLの最小全長木のエッジの数とする。NDをCEに加えて、CEがLLより長い最短エッジのインデックスと

10

20

30

40

50

する。

【0156】17dm) 工程17d1)に戻る。

【0157】17dn) ランク値Rが以下のように求められる。

【0158】

【数1】

$$R = \frac{LL - MIN}{LL - SL} \times \frac{ND}{T} + \frac{T - ND - NH}{T} \quad (1)$$

【0159】ランクの推定 - 楕円面の場合

ここで、超楕円面境界が定義される第2の整列の場合の補間によるランク値の計算について以下で詳細に説明する。図8で示される各サンプルから楕円面境界を定義する代わりに、エッジの両端の複数のサンプルを焦点として楕円面境界を定義する。工程17が以下のような工程17'によって置き換えられる。

【0160】17a') 現在のクラスタとテストサンプルに限定して、サンプルの最近傍のものを探索する工程13a)と同様の方法を使って、テストサンプルの最近傍のクラスタ内でサンプルNSPを見つける。

【0161】17b') テストサンプルからクラスタ内で最近傍のサンプルNSPまでの距離dとクラスタ内の最大エッジ長eを求める。もしクラスタに1つのサンプルだけが含まれエッジがないなら、 $e = d/2$ とする。もし $d > (1.5 \times e)$ なら、テストサンプルは完全にクラスタの外にあると決定され、また、ランク値が以下のような値に割り当てられる。

【0162】17ba') 探索の指示と制限を行うためにインデックスが付けられた絶対距離を使って、テストサンプルへの最小ストリング距離Sをもつクラスタの最小全長木内のエッジを見つける。エッジによって接続された2つのサンプルをsource(i)とdest(i)配列から見つけることができる。もし現在の最小のSがrなら、より小さいSのエッジは、テストサンプルの $1.5 \times r$ の絶対距離内に存在する少なくとも1つのサンプルをもつはずである。従って、この範囲に探索が制限される。dは最小の値Sとなり、また、 $R = 1 - e/d$ である。そして、本処理が停止する。

【0163】17c') テストサンプルがクラスタに存在する。もしクラスタに単一のサンプルだけが含まれるなら、 $R = 1 - d/e$ として停止する。

【0164】17d') クラスタには複数のサンプルが含まれる。以下の工程を実行する。

【0165】17da') Tをクラスタの最小全長木のエッジ数とする。

【0166】17db') 複数のエッジに対してその長さの昇順にインデックスを付けることを考慮して、CEを現在のエッジのインデックスとする。dと(r-d)のうちの大きい方の2/3以上の高さの第1のエッジにCEを初期設定する。ここで、rはNSPの併合の高さである。

【0167】17dc') NLEを、エッジCEより長い第1のエッジのインデックスとする。

【0168】17dd') NHをエッジCEより短いエッジの数とする。

【0169】17de') MINを、テストサンプルから、これまで考えられた中で最近傍のサンプルまでの距離とする。MINは最大値に初期的設定される。

【0170】17df') SLを、エッジCEより短い、MSTで最長のエッジの長さとする。

【0171】17dg') LLを現在のエッジCEの長さとする。LLはゼロに初期設定される。

【0172】17dh') NDを、LLに等しい長さのクラスタの最小全長木内のエッジの数とする。NDは1に初期設定される。

【0173】17di') 現在のエッジCEに対するLLを求める。

【0174】17dj') NDを求める。ND分CEをインクリメントすることによって、CEは現在、LLより長い最短エッジのインデックスとなる。

【0175】17dk') 'アクティブエッジ'AEの群を、サンプル初期的にサンプルがないものとして定義する。'アクティブ領域'ARは、AEのエッジの長さLLの'ストリング距離'S内の全サンプル領域として定義される。'ストリング距離'は、テストサンプルから、エッジによって接続される各サンプルまでの距離の合計値SDとエッジの長さの差の1/2である。即ち、 $S = (SD - LL) / 2$

17d1') 更新されたARを見つけ、テストサンプルがその中に入るかどうかを以下のようにチェックする。

【0176】17d1a') 併合の高さLLの複数のサンプルに複数のアクティブサンプルASを加える。併合の高さLLの各サンプルに対して、テストサンプルに対するストリング距離を測定する。もしSがMINより小さければ、SとSLの大きい方をMINに設定する。

【0177】17d1b') もしMINがLL以下であれば、工程17dn')に進む。ここで、テストサンプルがアクティブ領域にあるときのランク値を求める。

【0178】17d1c') $SL = LL$ に設定し、LLを新たなエッジの長さとする。LLより短いエッジの数がND分増えたとき、NDをNHに加える。NDに長さLLの最小全長木のエッジの数を設定する。NDをCEに加えることによって、CEはLLより長い最短エッジのインデックスとなる。もし最小全長木にエッジがもはや存在しなければテストサンプルはクラスタの外側にあるので、17dn')に進む。

【0179】17dm') 17d1')に進む。

【0180】17dn') 推定ランク値は、

【0181】

【数2】

$$R = \frac{LL - MIN}{LL - SL} \times \frac{ND}{T} + \frac{T - ND - NH}{T} \quad (1')$$

によって与えられる。

【0182】17dO') 負の推定ランク値は、 $R = MIN - SL$ によって与えられる。

【0183】分類

1つより多いクラスタがある場合は、テストサンプルが最も大きな推定ランク値をもつクラスタに割り当てられる。本処理には、画像や音声等の入力信号の自動認識を含む一般的な人工知能分野で多くの実用的アプリケーションがある。例えば、音声認識のアプリケーションでは、訓練の段階で一連の訓練サンプルがデータ入力部Iに入力される。何の音、もしくは、何の音素を訓練サンプルが意図的に表すかは周知の先験的事項であるので、意図した分類に基づいて複数のサンプルが分類され、各分類の複数のサンプルが独立にクラスタリングされる。認識モードでは、各クラスタに関連するテストサンプルに対してランク値を計算し、これらのランク値間の比較に基づいてテストサンプルを分類する。1つの分類だけをテストサンプルに割り当てることによって'ハード'的な分類を行うことができる。あるいは、テストサンプルが属する確率を複数の様々な分類の各々に割り当てることによって'ソフト'的な分類を行うことができる。'ソフト'的な分類を使って、各音の'ソフト'的な分類確率によって重み付けられた一連の音の相対確率を決定することによって、一連の音を分類することができる。

【0184】本技術は、複数のクラスタの包絡線が要素空間の他のクラスタのものと部分的にオーバーラップしても、分類の決定を行うことができるという利点をもつ。

【0185】本技術を複数のデータサンプルに適用することができる。ここで、各データサンプルは、例えば、認識された文字の示値が出力値となる光学的文字認識(OCR)、もしくは、テストサンプルの分類に対応して処理される1つ以上の活動が出力値であるロボットビジョンの分野での空間的構成を表すものである。

【0186】従って、本処理を実行する装置のデータ入力部Iはセンサ、もしくは、センサアレイであり、その装置の次の処理工程に影響を与える入力値の分類をデータ出力部Oが示す。

【0187】特定の例

ここで、テストサンプルのランク値を推定する特定の例を図9を参照して説明する。サンプル4、6間の最長のエッジを除去することによって、図8の単一のクラスタを2つのクラスタに分割する。これは、4.12と4.47間の高さで二進木をカットすることとして表現される。2つのクラスタのどちらにテストサンプルが属するべきかを決定するために、座標(6, 3)にテストサンプルTPを与えて、各クラスタに対するテストサンプルのランク値を求める必要がある。ここで、ランクの輪郭を図9で示す。これと比較して、長楕円の場合のランク

28

の輪郭を図10で示す。

【0188】以下で説明される超長球面の場合では各クラスタを順番に選ぶ。サンプル(1, 2, 3, 4)のクラスタではテストサンプルの最近傍のサンプルはサンプル4であり、これに対してdは $\sqrt{5}$ である。eが $\sqrt{10}$ なので、TPは本クラスタ内にある。昇順の長さb、a、cの3つのエッジがある。CEはエッジaを初期的指す。何故ならば、これはdの長さに等しく、また、 $LL = \sqrt{5}$ であるからである。サンプル1, 2, 3を複数のアクティブなサンプルに加える。サンプル3はTPに近いので、MINは $\sqrt{17}$ となる。これはLLより大きいので、次に長いエッジである長さ $\sqrt{5}$ のエッジを見つける。ここで、工程17d1)に戻り、サンプル1を複数のアクティブサンプルに加える。しかしながら、サンプル1は既に検査されたサンプルよりもさらにTPから離れているので、MINはまだ $\sqrt{17}$ である。これはLLより大きい。従って、次に長いエッジ、即ち、エッジcを見つける。サンプル4を複数のアクティブサンプルに加え、 $LL = \sqrt{10}$ とするが、 $SL = \sqrt{5}$ である。MINは $\sqrt{5}$ になり、これはLLより小さい。従って、TPが $R = 1/3$ の境界に存在することを観察することによって確認可能な

【0189】

【数3】

$$R = \frac{\sqrt{10} - \sqrt{5}}{\sqrt{10} - \sqrt{5}} \times \frac{1}{3} + \frac{3 - 1 - 2}{3} = \frac{1}{3} \quad (2)$$

を計算する。

【0190】上の実施形態はユークリッド測定基準で説明されたが、その他の種類の測定基準もその代わりに使うことができることを評価すべきである。さらに、本発明の複数の態様を、非測定基準のデータサンプルと、単一リンクのクラスタリング以外のクラスタリング方法に適用することができる。

【図面の簡単な説明】

【図1】図1は本発明の実施形態に係る方法の主要な工程を示すフローチャートである。

【図2】図2は図1フローチャートの工程13aの詳細な工程を示すフローチャートである。

【図3】図3は図1、図2で示される方法を利用して演算された一連のデータサンプルの最小全長木を示す。

【図4】図4は最小全長木を計算する途中の二進木を示す。

【図5】図5は演算の最終段階の二進木を示す。

【図6】図6は、ある技術的な処理方法を実行する装置の一般図である。

【図7】図7は併合の高さの順に並べられた図5の二進木を示す。

【図8】図8は、図3のデータサンプルのクラスタに対するランク値関数の超球の輪郭を示す。

【図9】図9は、図3のデータサンプルのサブクラスタ

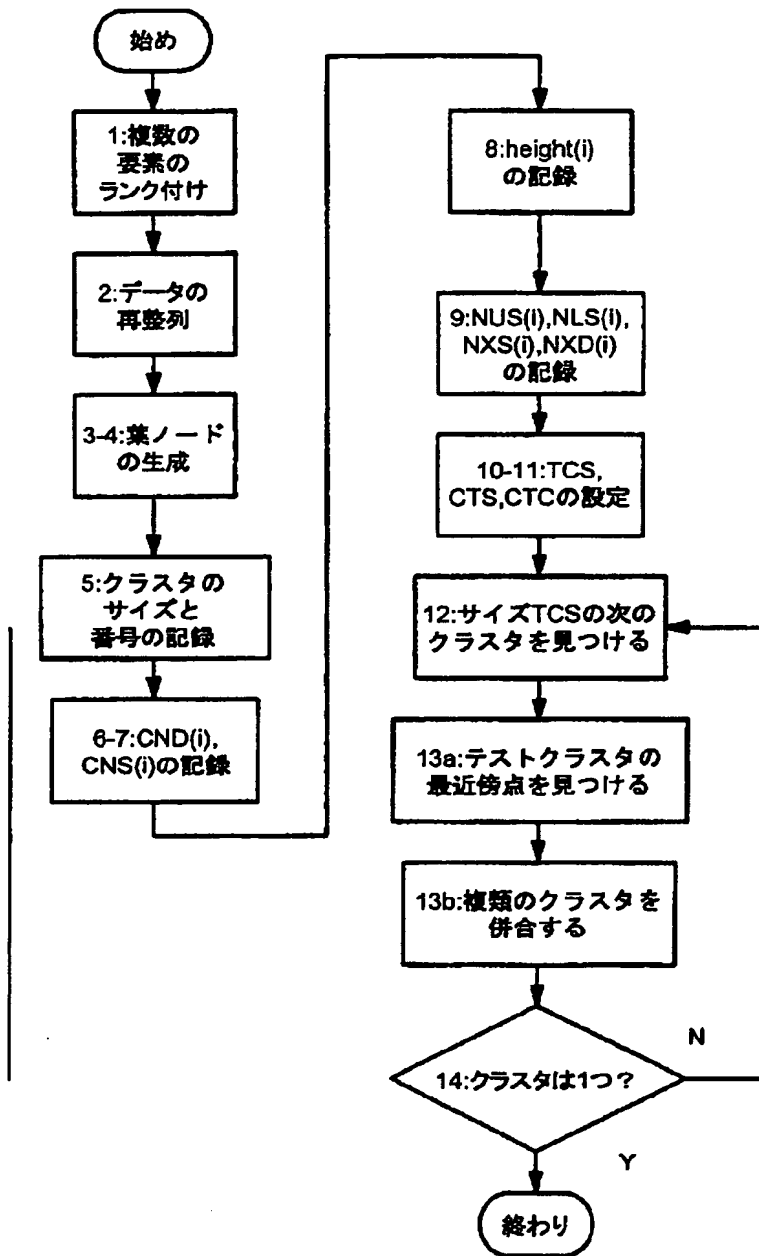
に対するランク値関数の超球の輪郭を示す。

【図10】図10は、図9のサブクラスタに対するランク値関数の超球の輪郭を示す。

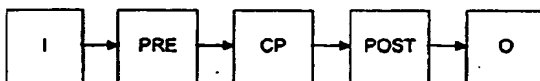
*

*【図11】図11はランク値関数演算のフロー図である。

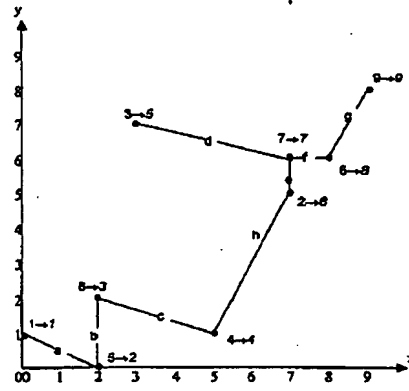
【図1】



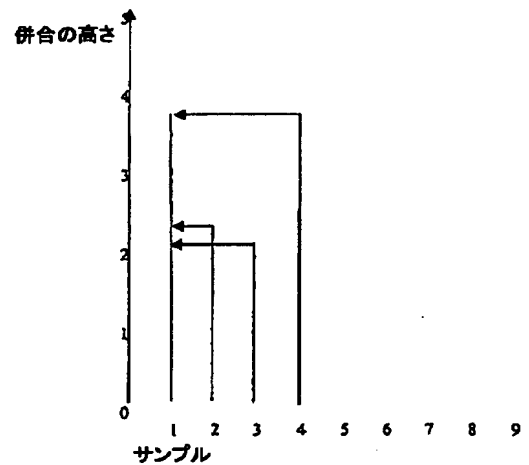
【図6】



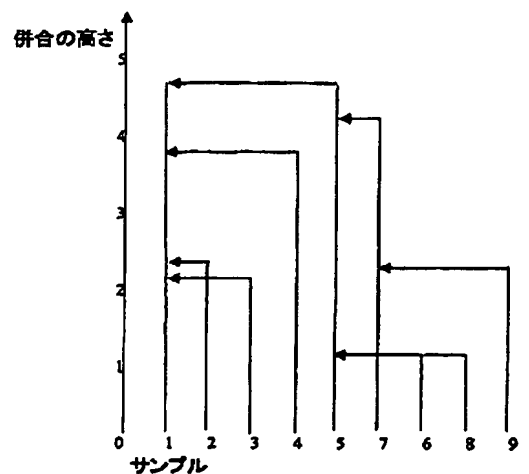
【図3】



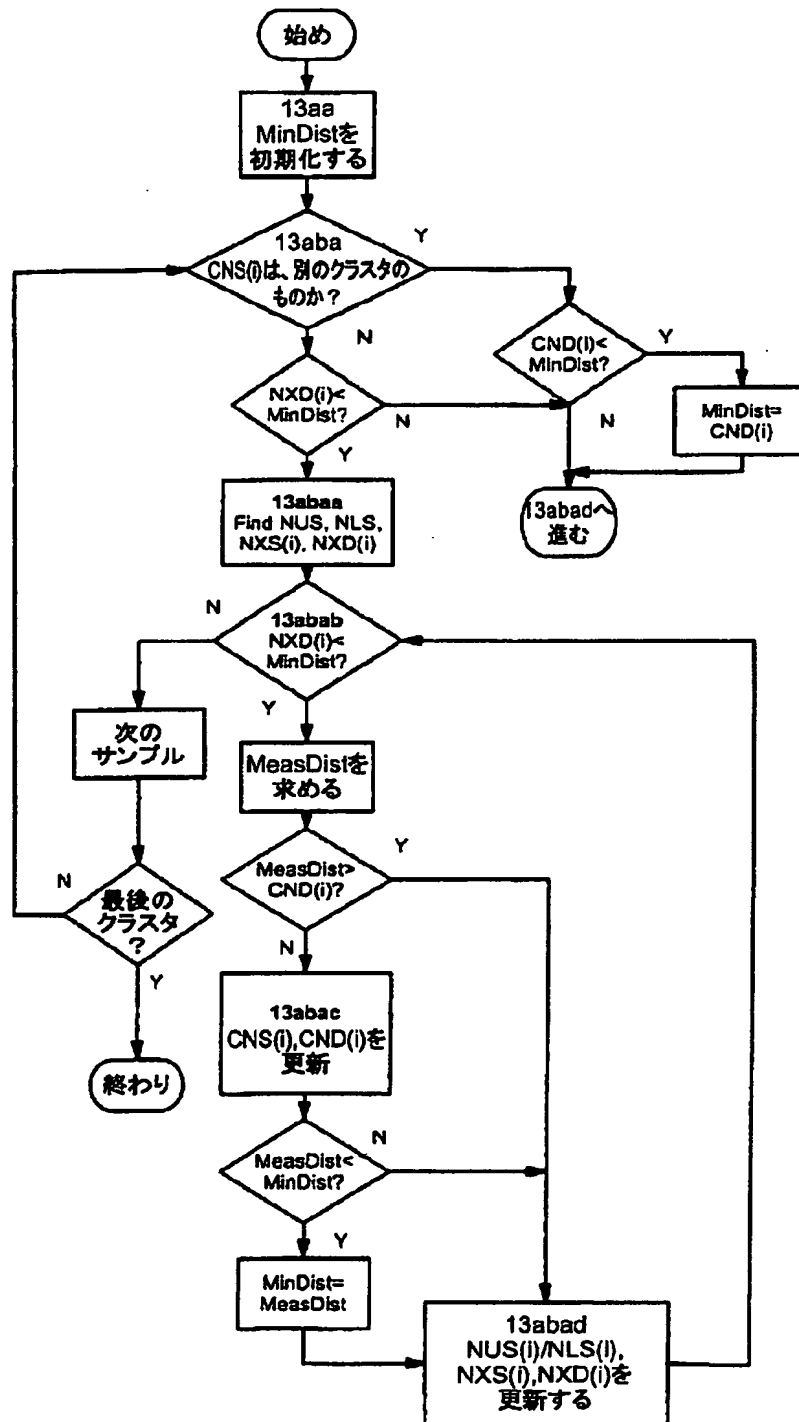
【図4】



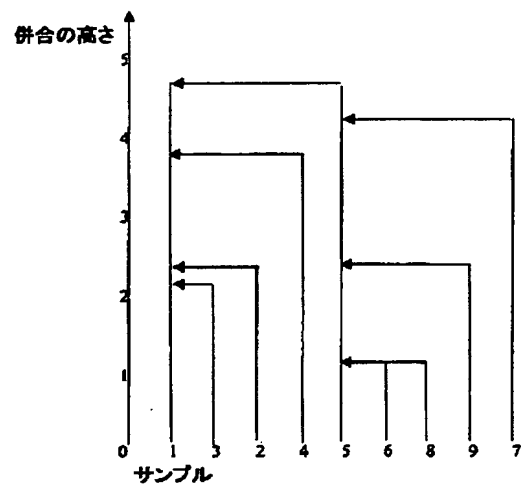
【図5】



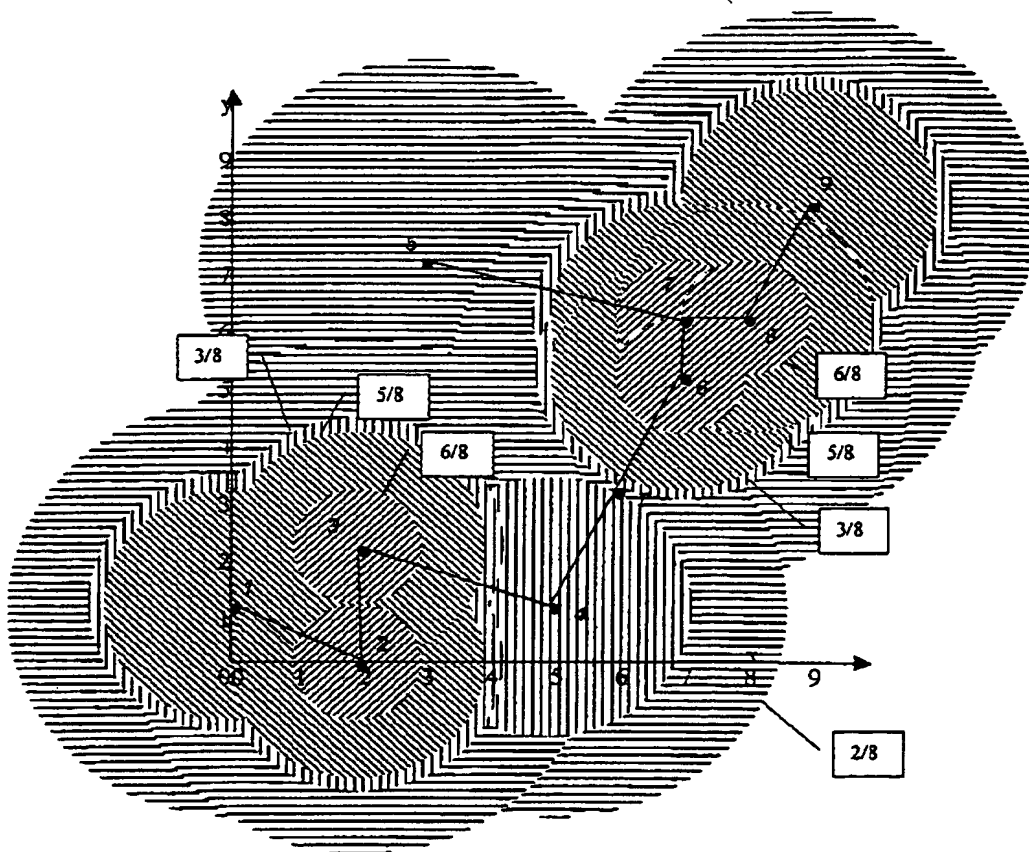
【図2】



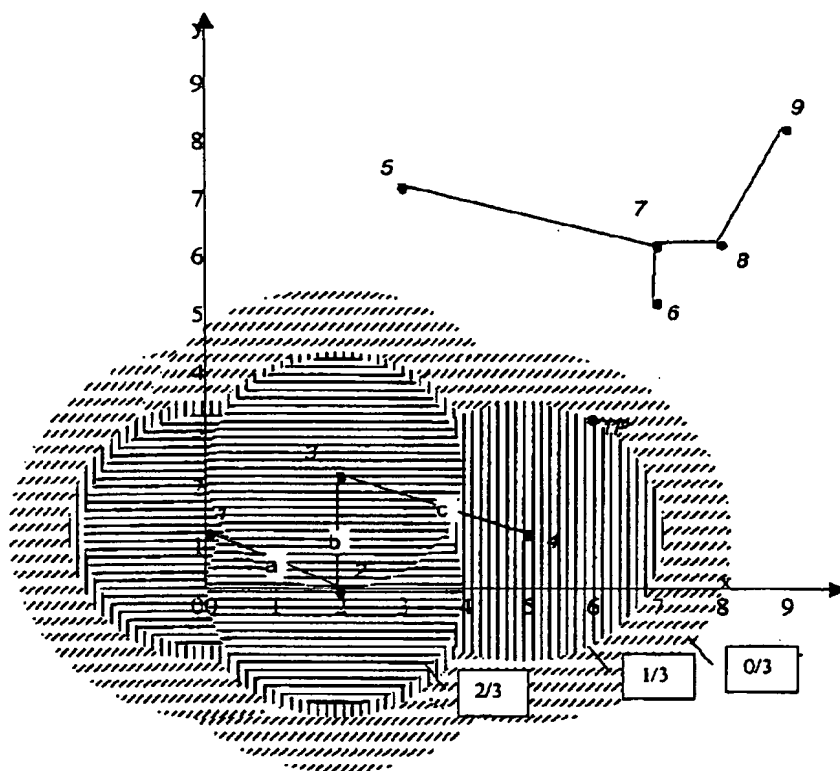
【図7】



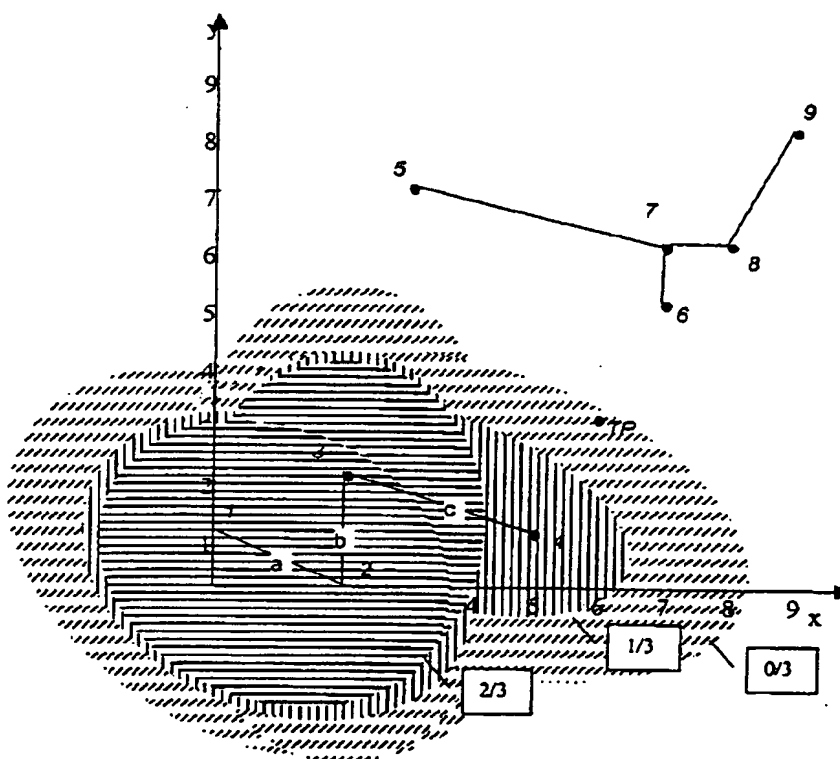
【図8】



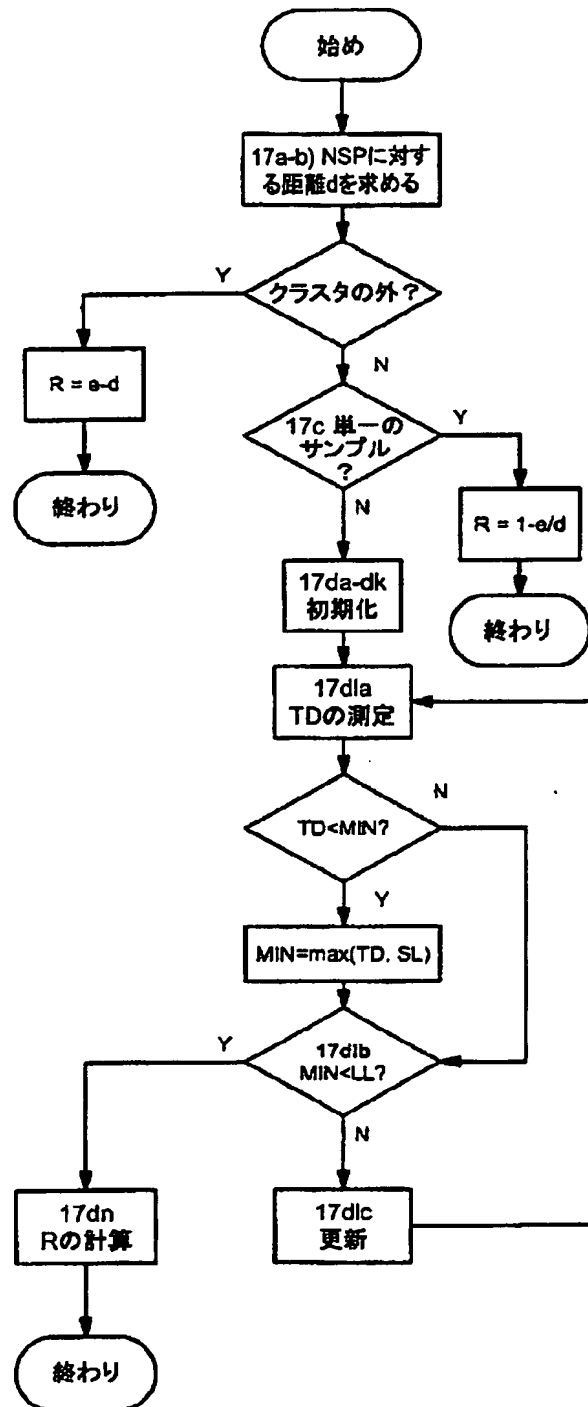
【図9】



【図10】



【図11】



フロントページの続き

F ターム(参考) 5B075 ND03 ND06 ND18 ND23 ND35
ND40 NR02 NR03 NR12 NR15
NR16 NS01 PP30 PQ05 PQ36
PR06 QM08 QS11 UU29 UU40
5L096 FA66 JA03 MA07

【外国語明細書】

1 Title of Invention

Data Clustering Methods and Applications

2 Claims

1. A hierarchical data clustering method, including the following steps:
 - a. receiving as input a set of data samples;
 - b. recording an initial cluster allocation of said data samples;
 - c. for each cluster of said data samples, determining the most similar cluster to that cluster and recording the dissimilarity thereto, according to a predefined dissimilarity function;
 - d. recording the identities of the data samples from which the dissimilarity was determined;
 - e. recording the most similar cluster and the current cluster as a single cluster;
 - f. repeating steps c to e until a predetermined degree of clustering is reached;
 - and
 - g. providing as output the recorded dissimilarities and associated data sample identities;wherein at step c, the clusters are taken in order of increasing size.
2. A method according to claim 1, wherein at step c, the cluster having the most similar data sample relative to any of the data samples within the current cluster is determined as the most similar cluster.
3. A hierarchical data clustering method, including the following steps:
 - a. receiving as input a set of data samples;
 - b. indexing said set of data samples substantially in order of absolute distance from a common reference, according to a predefined absolute distance metric;
 - c. recording an initial cluster allocation of said data samples;
 - d. for each cluster of said data samples, determining the closest cluster to the current cluster and recording the distance thereto, according to a predefined intersample distance metric;

- e. recording the identities of the data samples from which the distance was determined;
 - f. recording the closest cluster and the current cluster as a single cluster;
 - g. repeating steps d to f until a predetermined degree of clustering is reached; and
 - h. providing as output the recorded distances and data sample identities; wherein step d includes selecting for distance comparison with each data sample within the current cluster only a subset of the data samples outside the current cluster within an index range between a higher index having an index value higher than that of the current data sample and a lower index having an index value lower than that of the current data sample.
4. A method according to claim 3, wherein step f includes, if the data samples of the closest cluster and the current cluster are not adjacent in index, reindexing at least some of the data samples so that the data samples of the closest cluster and the current cluster are adjacent.
5. A method according to claim 3 or claim 4, wherein the higher and lower indices are determined such that the smaller of the difference between the absolute distance of the data sample of the lower index and that of the current data sample, and the difference between the absolute distance of the data sample of the higher index and that of the current data sample, is greater than the minimum intersample distance between the current data sample and any of the data samples within the index range and not in the current test cluster.
6. A method according to claim 5, wherein the higher and lower indices are determined by successively reducing the lower index or increasing the higher index according to whether the difference between the absolute distance of the data sample of the lower index and that of the current data sample is respectively less than or greater than the difference between the absolute distance of the data sample of the higher index and that of the current data sample, until the smaller of the

differences is greater than the minimum intersample distance between the current data sample and any of the data samples within the index range and not in the current test cluster.

7. A method according to any one of claims 3 to 6, wherein the absolute distance metric is the difference in component in the dimension of the data samples having the greatest variation.
8. A method according to claim 7, wherein the variation is determined as the range within which a predetermined fraction of the data samples fall.
9. A hierarchical data clustering method including the following steps:
 - a. receiving as input a set of data samples each having a plurality of dimensions;
 - b. determining for each of said dimensions a measure of variation of the data samples in that dimension;
 - c. sorting the dimensions of the data samples according to their measures of variation;
 - d. setting initially each data sample as belonging to its own cluster;
 - e. taking each cluster in turn, determining the closest data sample to any sample in that cluster and not already forming part of that cluster;
 - f. merging the cluster of the closest data sample with the current cluster; and
 - g. repeating steps e) and f) until a desired degree of clustering has been achieved;

wherein the measure of variation is the range of a predetermined fraction of the data samples excluding the largest and smallest values in that dimension.

10. A hierarchical data clustering method, including the following steps:
 - a. receiving as input a set of data samples;
 - b. recording an initial cluster allocation of said data samples;

- c. for each cluster of said data samples, determining the most similar cluster to the current cluster and recording the dissimilarity thereof, according to a predefined dissimilarity function of a plurality of dimensions of the data samples;
 - d. recording the identities of the data samples from which the dissimilarity was determined;
 - e. recording the most similar cluster and the current cluster as a single cluster;
 - f. repeating steps c to e until a predetermined degree of clustering is reached; and
 - g. providing as output the recorded dissimilarities and associated data sample identities;
- wherein step c includes, for each dissimilarity calculation, taking the component of the distance measurement in each dimension in order of decreasing variation of data samples within each dimension, calculating a cumulative dissimilarity value, and terminating the dissimilarity calculation if the cumulative dissimilarity value exceeds a comparative dissimilarity value.

- 11. A data compression method, including:
 - a. performing a method according to any preceding claim;
 - b. generating a compressed data set based on the set of data samples and the output of the method of step a; and
 - c. outputting said compressed data set.
- 12. A method according to claim 11, including storing said compressed data set.
- 13. A method according to claim 11, including transmitting said compressed data set.
- 14. A feature extraction method, including:
 - a. performing as method according to any of claims 1 to 10; and

- b. indicating associations between ones of said data samples within the same cluster on the basis of the output of the method of step a.
- 15. A method according to claim 14, wherein step b includes comparing properties of at least one of the data samples with predetermined classification data according to the clustering properties of the at least one data samples, and outputting a classification indication of the data samples within the same cluster of the basis of the comparison.
- 16. An unmixing method, including:
 - a. performing a method according to any one of claims 1 to 10;
 - b. determining at least two characteristic properties of ones of said data samples on the basis of the clustering properties of said data samples determined by step a; and
 - c. determining a mixing proportion of the characteristic properties for at least one of the data samples.
- 17. A method according to claim 16, including determining at least one boundary region between ones of the data samples having one of the characteristic properties; wherein in step c, the at least one data samples are within the boundary region.
- 18. A method according to claim 16, wherein the boundary region is determined by spatial or temporal edge detection.
- 19. A method according to claim 17 or 18, including indicating as an anomaly any of the data samples within the boundary region of which the value is determined not to consist of a mixing proportion of the characteristic properties.
- 20. A method of data selection, including:
 - a. performing as method according to any of claims 1 to 10; and

- b. selecting for further processing a subset of said data samples on the basis of their clustering properties as determined at step a.
- 21. A method according to claim 20, wherein said subset is selected to comprise a cluster in accordance with predefined clustering criteria within said cluster.
- 22. A method according to claim 20, wherein said subset is selected to comprise cluster in accordance with predefined clustering criteria relative to other clusters.
- 23. A method according to claim 20, including pre-selecting at least one of said data samples, wherein the subset is selected to comprise a cluster including the or each pre-selected data sample.
- 24. A method according to claim 20, including pre-selecting at least one of said data samples, wherein the subset is selected to comprise a cluster excluding the or each pre-selected data sample.
- 25. A method of generating a network design, including the steps of:
 - a. performing a method as claimed in any one of claims 1 to 10, wherein the data samples represent nodes of a network; and
 - b. generating a representation of interconnections between the nodes of the network in accordance with a minimum spanning tree defined by the output of step a.
- 26. A method of network construction, including the steps of:
 - a. performing a method as claimed in any one of claims 1 to 10, wherein the data samples represent nodes of a network; and
 - b. creating interconnections between the nodes in accordance with a minimum spanning tree defined by the output of step a.

- 27.** A method of classifying a test sample relative to a cluster comprising a plurality of data samples, including the steps of:
- a. determining the most similar data sample of the cluster to the test sample;
 - b. calculating a value associated with the test sample and the cluster, dependent on the dissimilarity of the test sample to the most similar data sample and the dissimilarity of the most similar data sample to any other data sample within the cluster; and
 - c. performing further processing steps dependent on the calculated value for the cluster;
- wherein at step b, the value is calculated as a function of the dissimilarity of the test sample to the most similar data sample and of the dissimilarity of the test sample to another data sample most similar to the most similar data sample within the cluster.
- 28.** A method of classifying a test sample relative to a cluster comprising at least three data samples, including the steps of:
- a. calculating a value associated with the test sample and the cluster, dependent on the dissimilarity between pairs of data samples within the cluster; and
 - b. performing further processing steps dependent on the calculated value for the cluster;
- wherein at step a, the value is calculated as a function of a test sample dissimilarity of the test sample to the most similar data sample within the cluster, unless the test sample dissimilarity is less than the dissimilarity of an edge in a minimum spanning tree which has the greatest dissimilarity less than an edge connected to the most similar data sample.
- 29.** A method of according to claim 27 or 28, including calculating a value associated with the test sample relative to each of one or more further clusters; wherein said further processing steps are performed on the basis of a comparison

between the values calculated for each of the clusters.

30. A pattern recognition method, including:
 - a. receiving a test sample; and
 - b. performing a method according to any one of claims 27 to 29.
31. A pattern recognition method, including:
 - a. receiving a test sample;
 - b. performing a method according to any one of claims 1 to 10; and
 - c. performing a method according to any one of claims 27 to 29.
32. A method according to any preceding claim, wherein the data samples are samples of physical properties.
33. A computer program arranged to perform a method according to any preceding claim when executed by a suitably arranged processor.
34. A carrier carrying a computer program according to claim 33.
35. Apparatus arranged to perform a method according to any one of claims 1 to 32.
36. Apparatus comprising a data input, a pre-processing stage, a cluster processor arranged to perform a method according to any of claims 1 to 10, a post-processing stage and a data output.

3 Detailed Description of Invention

Field of the Invention

The present invention relates in one aspect to a hierarchical data clustering method and to processes and applications including that method. In another aspect, the present invention relates to a method of classifying a test sample relative to one or more clusters. The present invention relates further to processes and applications involving such methods.

Background of the Invention

Hierarchical cluster analysis involves the classification of a set of data samples into a cluster structure based on the similarity between the samples, without imposing any predefined grouping. There is a large body of literature on methods and applications of cluster analysis, examples of which are:

'Data Clustering: A Review' by Jain A. K., Murty, M. N. and Flynn, P. J., ACM Computing Surveys No. 3, vol. 31, p. 264;

'Classification', by Gordon, A. D., Chapter 3, published Chapman & Hall, 1981.

Single-link cluster analysis is one type of cluster analysis which involves finding the 'minimum spanning tree' of a set of data samples. The 'minimum spanning tree' is a set of lines or 'edges' which join pairs of data samples such that the total length or 'weight' of the edges is a minimum; see for example:

'Introduction to Algorithms', by Cormen, T. E, Leiserson, C. E., and Rivest, R. L., Chapter 24, published MIT Press 1990;

'Algorithms', by Sedgewick, R., Chapter 31, Second edition 1988, published by Addison Wesley.

In practical applications, the usefulness of any cluster analysis technique depends on its efficiency in terms of speed and storage requirements, which will be functions of the number of data samples N , the number of dimensions D of each sample, and the structure of the data samples. In the worst case, hierarchical clustering algorithms have time requirements of the order of N^2 and can therefore be impractical for large data sets.

The paper 'An Efficient Interactive Agglomerative Hierarchical Clustering Algorithm for Hyperspectral Image Processing' by the present inventor, published in the

Proceedings of the SPIE Conference on Imaging Spectrometry, San Diego, California, July 1998, SPIE Vol. 3438, pp. 210 to 221 describes clustering algorithms which involve indexing data points so that nearby points have nearby indices and searching in a restricted subspace so as to reduce the number of comparisons which need to be made between pairs of points. However, the search is made only in one direction and does not achieve accurate single-link clustering.

European patent publication EP 913 780 A discloses a data clustering method in which the total number of distance calculations is reduced by eliminating data samples unlikely to be nearest to the sample under consideration before selecting the nearest.

The paper 'A Spectral Unmixing Algorithm for Distributed Endmembers with Applications to BioMedical Imaging', Proceedings of SPIE, Vol. 3438, by the inventor of the present invention, discloses a method of calculating a likelihood value of a test point belonging to a set of data points, based on a hierarchical clustering of the data points.

Summary of the Invention

According to one aspect of the present invention, there is provided a data clustering method including the following steps:

- a. Receiving as input a set of data samples;
- b. Setting initially each data sample as belonging to its own cluster;
- c. Taking each cluster in turn, determining the closest data sample to any sample in that cluster and not already forming part of that cluster;
- d. Merging the cluster of the closest data sample with the current cluster; and
- e. Repeating steps c and d until a desired degree of clustering has been achieved; wherein at step c the clusters are taken in order of increasing cluster size.

An advantage of this method is that the number of distance or dissimilarity measurements which need to be calculated is greatly reduced, because there are generally fewer distances between samples in the smallest current cluster and another cluster of larger size than between samples in clusters both of larger size than the current smallest cluster.

According to another aspect of the present invention, there is provided a data clustering method including the following steps:

- a. Receiving as input a set of data samples;
- b. Setting initially each data sample as belonging to its own cluster;
- c. Taking each cluster in turn, determining the closest data sample to any sample in that cluster and not already forming part of that cluster;
- d. Merging the cluster of the closest data sample with the current cluster; and
- e. Repeating steps c and d until a desired degree of clustering has been achieved; wherein at step c, the closest data sample is determined by searching over a restricted subspace defined by an index range of the data samples indexed according to distance from a reference.

An advantage of this method is that, by indexing the data samples according to distance from a reference, there will be a maximum and minimum index relative to a sample under consideration within which the nearest sample must be contained and the number of samples which need to be compared to the sample under consideration is greatly reduced, without compromising the accuracy of the clustering.

According to another aspect of the present invention, there is provided a data clustering method including the following steps:

- a. Receiving as input a set of data samples each having a plurality of dimensions;
- b. Determining for each of said dimensions a measure of variation of the data samples in that dimension;
- c. Sorting the dimensions of the data samples according to their measures of variation;
- d. Setting initially each data sample as belonging to its own cluster;
- e. Taking each cluster in turn, determining the closest data sample to any sample in that cluster and not already forming part of that cluster;
- f. Merging the cluster of the closest data sample with the current cluster; and
- g. Repeating steps e) and f) until a desired degree of clustering has been achieved; wherein the measure of variation is the range of a predetermined fraction of the data samples excluding the largest and smallest values in that dimension.

An advantage of this method is that the dimensions which are most likely to be of significance in determining dissimilarities between samples are considered first and it is

therefore often unnecessary when making comparisons between samples to consider all of the dimensions.

According to another aspect of the present invention, there is provided a method of classifying a test sample relative to a cluster comprising a plurality of data samples, including the steps of:

- a. determining the most similar data sample of the cluster to the test sample;
 - b. calculating a value associated with the test sample and the cluster, dependent on the dissimilarity of the test sample to the most similar data sample and the dissimilarity of the most similar data sample to any other data sample within the cluster; and
 - c. performing further processing steps dependent on the calculated value for the cluster;
- wherein at step b, the value is calculated as a function of the dissimilarity of the test sample to the most similar data sample and of the dissimilarity of the test sample to another data sample most similar to the most similar data sample within the cluster.

An advantage of this method is that the calculated value is calculated with reference to an edge rather than an individual sample and provides a smoother variation in value in regions intermediate the samples joined by the edge.

According to another aspect of the present invention, there is provided a method of classifying a test sample relative to a cluster comprising at least three data samples, including the steps of:

- a. calculating a value associated with the test sample and the cluster, dependent on the dissimilarity between pairs of data samples within the cluster; and
- b. performing further processing steps dependent on the calculated value for the cluster;

wherein at step a, the value is calculated as a function of a test sample dissimilarity of the test sample to the most similar data sample within the cluster, unless the test sample dissimilarity is less than the dissimilarity of an edge in a minimum spanning

tree which has the greatest dissimilarity less than an edge connected to the most similar data sample.

An advantage of this method is that the test sample is given greater weight when close to the shortest edge than when close to the next shortest edge and therefore not as close to the tightest region of the cluster.

Description of Specific Embodiments

A method according to one embodiment of the invention is described below. An array of N samples each of D dimensions is provided as input.

Step 1 – Rank Dimensions

The interquartile range, in other words the range between the first and third quartiles and hence containing the middle 50% of the samples, is calculated for each of the D dimensions. The interquartile range is an advantageous measure in this case, because it is little affected by stray samples at the extremes of the range and therefore gives a good representation of the variation for the majority of the samples. The dimensions of the sample array are reordered in order of decreasing interquartile range, or a ranking order of the dimensions is stored in a dimension rank array.

Step 2 – Reorder Data Samples

The data samples are reordered within the array in one of two ways:

2a) Radial Ordering: the samples are reordered in order of increasing distance from an origin, which is selected for example to be the sample with the smallest component in the dimension with the largest interquartile range.

2b) Linear Ordering: the samples are reordered in order of increasing value in a selected dimension (preferably the dimension with the largest interquartile range). In both cases, the original index is stored in a one-dimensional array so as to allow identification of individual samples.

Step 3 – Create Binary Tree Leaf Nodes

Each sample is assigned to a corresponding leaf node of a binary tree so that the i^{th} sample after reordering is assigned initially to the i^{th} leaf node of the binary tree, and the assignment is stored in an array.

Step 4 – Create Cluster Labels

An array of cluster labels is created indicating the cluster to which each sample belongs. Each sample is initially considered to belong to its own cluster and hence the cluster label is initially the index number of the sample.

Step 5 – Record Cluster Size and Number

The size (number of samples) of each cluster and the number of clusters (initially N) is recorded.

Step 6 – Record Nearest Distance

The distance from each unmerged sample to the nearest sample in a different cluster is stored as a variable $CND(i)$ (Current Nearest Distance).

Step 7 – Record Nearest Sample

The index of the nearest sample is stored as an integer $CNS(i)$ (Current Nearest Sample).

Step 8 – Record Merge Height

The distance of each unmerged sample to the cluster to which it is merged is stored as the 'merge height'. Initially, no merging has been done so that the distance is set as infinity (i.e. a maximum value).

Step 9 – Record Inter-sample and Next Distances

For each sample, the samples are found having the next highest and lowest leaf node indices not within the same cluster; these will be referred to as the 'next upper' and 'next lower' samples $NUS(i)$, $NLS(i)$ respectively. For example, if the test sample is at leaf node i , initially the next upper sample $NUS(i)$ will be at leaf node $i+1$ and the next lower sample $NLS(i)$ at leaf node $i-1$, because each sample initially belongs only to its own cluster.

For each sample and its next upper and lower samples, the 'absolute distance' from the origin is calculated. If radial ordering was used at step 2a), then the 'absolute distance' is the radial distance from the chosen origin. If linear ordering was used at step 2b), then the 'absolute distance' is the distance in the chosen direction from the chosen origin, which is preferably the sample with the smallest component in the chosen direction.

Next, the difference between the absolute distance of the sample and the absolute distance of the next upper sample $NUS(i)$, and the difference between the absolute distance of the sample and the absolute distance of the next lower sample $NLS(i)$ are calculated. The smaller of these two differences is stored as the 'next distance' $NXD(i)$ of the sample and the sample index of the next upper or next lower sample $NUS(i)$, $NLS(i)$ which gave the smaller of the two differences is stored as the 'next sample' $NXS(i)$.

Step 10 – Set Test Cluster Size

Test Cluster Size TCS is set initially to 1.

Step 11 – Set Current Test Sample and Cluster

The current test sample CTS is set initially to sample index 1, and the current test cluster CTC is the cluster containing that sample (initially cluster 1).

Step 12 – Find Next Cluster of Test Cluster Size

The clusters are examined in the order in which they appear in the binary tree to find the next cluster having a size equal to the test cluster size TCS. This is done as follows:

12a) If the current cluster size is equal to the test cluster size, then step 12 is complete.

12b) Otherwise, jump to the sample at the next leaf node immediately following the current test cluster, which will always be grouped in consecutive leaf nodes in the tree. Make the cluster containing this sample the current test cluster and go to step 12a). If the last leaf node has already been reached, set the test cluster size as the size of the smallest cluster present, as follows:

12ba) Increment the current test cluster size and set the 'current minimum test cluster size' to N (i.e. the maximum possible cluster size).

12bb) Set the current test sample to be the sample at the first leaf node.

12bc) If the size of the current test cluster is the same as the current test cluster size, then go to step 13.

12bd) If the size of the current test cluster is less than the current minimum test cluster size, update the current minimum test cluster size and store the current test cluster as the minimum size test cluster.

12be) If there are no more samples in the tree after the current test cluster, go to step 12bf). Otherwise, jump to the sample in the tree immediately following the current test cluster, make this the current test sample, and go to step 12bc).

12bf) Make the current test cluster the minimum size test cluster.

Step 13 – Merge with Nearest Sample Not Contained in Cluster

13a) The nearest sample to the current test cluster not itself contained in the test cluster is found as follows:

13aa) Set Minimum Distance MinDist as infinity (i.e. a maximum value)

13ab) For each sample in the current test cluster, do the following:

13aba) If the current nearest sample CNS(i) is not a member of the current test cluster CTC(i), and if the current nearest distance CND(i) is less than the minimum distance MinDist, set the minimum distance MinDist to be the current nearest distance CND(i), store the current sample index as the 'head' and the current nearest sample CNS(i) as the 'tail'.

Otherwise, if the next distance NXD(i) is less than the minimum distance MinDist, update the current nearest sample CNS(i) and current nearest distance CND(i) by resetting the current nearest distance CND(i) to infinity and proceed as follows:

13abaa) Find the next sample NXS(i) and the next distance NXD(i) for the current sample as in step 9.

13abab) If the next distance NXD(i) is not less than the minimum distance, return to step 13ab) with the next sample in the cluster as the current sample. Otherwise, measure the distance MeasDist from the current sample to the next sample NXS(i). If MeasDist is greater than the nearest distance CND(i), then go to step 13abad). The comparison is performed by measuring the component of the distance in each dimension in the order determined at step 1, squaring the value of that component, and adding the

squared value to a sum of squared values. After each summing operation, the sum is compared with the square of the nearest distance $CND(i)$, and if greater, then the operation proceeds to step 13abad) without summing any more terms. This method of performing the comparison avoids unnecessary calculations and gives improved speed, particular if D is large.

13abac) If $MeasDist$ is less than the nearest distance $CND(i)$, update the nearest sample $CNS(i)$ and the nearest distance $CND(i)$ to be the next sample $NXS(i)$ and $MeasDist$ respectively. If furthermore $MeasDist$ is less than $MinDist$, update $MinDist$ to be $MeasDist$, store the current sample as the 'head' and the next sample $NXS(i)$ as the 'tail'.

13abad) Update the next upper or next lower sample $NUS(i)$, $NLS(i)$, according to whether the next sample $NXS(i)$ was the next upper or next lower sample. If the next sample $NXS(i)$ was the next upper sample $NUS(i)$, then update the next upper sample $NUS(i)$ to be the sample with the next higher leaf node index following the current next upper sample $NUS(i)$ not in the test cluster. If the next sample $NXS(i)$ was the next lower sample $NLS(i)$, then update the next lower sample $NLS(i)$ to be the sample with the next lower leaf node index before the current next lower sample $NLS(i)$ not in the current test cluster. Recalculate the next sample $NXS(i)$ and next distance $NXD(i)$ taking into account the new next upper/lower sample $NUS(i)/NLS(i)$; then go to 13abab).

Once the last sample in the current test cluster has been processed, go to step 13b.

13b) At this stage the 'head' and 'tail' are the samples to be joined in the minimal spanning tree, and the current test cluster which contains the head is merged with the cluster which contains the tail, with the merge height set to be the minimum distance. The cluster having the higher value label is added to the cluster having the lower value label. This is done as follows:

13ba) The leaf positions of samples in the binary tree are rearranged such that the smaller cluster and any samples between the smaller and the larger cluster are swapped in leaf node position so that the smaller and larger clusters become adjacent. The sample

indices stored against each leaf node are updated to reflect the swap. The cluster having the higher value label is added to cluster having the lower value label by assigning the lower cluster label to the samples of the higher cluster.

13bb) The head and tail sample indices, the minimum distance MinDist (which is equal to the merging height) and the lower cluster label are stored in an array respectively as source(i), dest(i), height(i) and join(i), where i is the higher cluster label.

13bc) The array element storing the size of the lower cluster is increased by the size of the higher cluster.

13bd) The number of clusters is decremented.

Step 14 – Repeat

If there is only one cluster left, end the procedure; otherwise go to step 12.

The array source(i), dest(i) and height(i) are sufficient to define the minimum spanning tree and binary tree of the samples. Join(i) provides redundant information which nevertheless saves subsequent processing steps.

Specific Example

An example of the above method will now be described with reference to Figures 3 to 5, in a simple example where the number of dimensions D is 2 and the number N of data samples or 'patterns' x is 9. Figure 3 shows the values of the data samples as follows:

$x(1)=(0,1)$; $x(2)=(7,5)$; $x(3)=(3,7)$; $x(4)=(5,1)$; $x(5)=(2,0)$; $x(6)=(8,6)$; $x(7)=(7,6)$; $x(8)=(2,2)$; $x(9)=(9,8)$.

At step 1), there is no need to reorder the dimensions as the interquartile range is the same for both x and y .

At step 2), the radial ordering method of step 2a) is used. The sample (0,1) is chosen as the origin and the samples are reordered (1, 5, 8, 4, 3, 2, 7, 6, 9) and re-indexed in their new order, shown in italics in Figure 3. If the linear ordering

method of step 2b) were used and the x dimension chosen, the order would be (1, 5, 8, 3, 4, 2, 7, 6, 9).

In steps 6, 7 and 9, the following values are obtained:

Sample Index:	1	2	3	4	5	6	7	8	9
CND(i):	2.24	2	2	3.61	4.12	1	1	1	2.24
CNS(i):	2	3	2	3	7	7	8	7	8
NXD(i):	2.24	2	2	3.61	4.47	1	1	1	2.24
NXS(i):	2	3	2	3	6	7	8	7	8

At step 12, sample 1 is selected as the current test sample and the test cluster label is also set to 1. At step 13, sample 2 is joined to sample 1 as shown by edge *a* in Figure 3, and cluster 2 is merged into cluster 1. The following information is recorded:

source(2)=1; dest(2)=2; height(2)=2.24; join(2)=1

Next, sample 3 becomes the current test sample at step 12 as it belongs to the next cluster. At step 13, sample 3 is joined to sample 2 as shown by edge *b* in Figure 3, and cluster 3 is merged into cluster 1. The following information is recorded:

source(3)=3; dest(3)=2; height(3)=2; join(3)=1

Next, sample 4 becomes the current test sample at step 12. At step 13, sample 4 is joined to sample 3 as shown by edge *c* in Figure 3, and cluster 4 is merged into cluster 1. The following information is recorded:

source(4)=4; dest(4)=3; height(4)=3.61; join(4)=1

Next, Sample 5 becomes the current test sample at step 12. At step 13, sample 5 is joined to sample 7, as shown by edge *d* in Figure 3, and cluster 7 is merged into cluster 5. This necessitates a swap in leaf node position between samples 6 and 7 in

step 13ba), so that the tree now appears as shown in Figure 3. The following information is recorded:

source(7)=5; dest(7)=7; height(7)=4.12; join(7)=5

Next, sample 6 becomes the current test sample because it is at the next leaf node. Sample 6 is joined to sample 7 as shown by edge *e* in Figure 3 and cluster 6 is merged into cluster 5 (of which sample 7 is already a member). The following information is recorded:

source(6)=6; dest(6)=7; height(6)=1; join(6)=5

Next, sample 8 becomes the current test sample and is joined to sample 7 as shown by edge *f*. Cluster 8 is merged into cluster 5 and the following information is recorded:

source(8)=8; dest(8)=7; height(8)=1; join(8)=5

Next, sample 9 becomes the current test sample and is joined to sample 8 as shown by edge *g*. Cluster 9 is merged into cluster 5 and the following information is recorded:

source(9)=9; dest(9)=8; height(9)=2.24; join(9)=5.

Now, the test cluster size increases to 4 in step 12, as this is the minimum size of cluster present, and cluster 1 is taken as the current test cluster. Samples 1 to 4 are taken in turn, in leaf node order.

With test sample 1, at step 13aba) the nearest sample $CNS(1)$ is sample 2, which is in the same cluster and the method proceeds to step 13abaa). As there is no next lower sample, the next upper sample $NUS(i)$ and the next sample $NXS(1)$ are both sample 5. The next distance $NXD(i) = \sqrt{45}$, as is the measured distance $MeasDist$. This is less than $MinDist$ and the nearest distance $CND(1)$, so $CND(1) = MinDist = MeasDist$ and $CNS(1)=5$. At step 13abad) the next upper sample $NUS(i)$ is incremented to sample 6 and the method returns to step 13abab. The next distance

$NXD(1)$ is the difference between the absolute distances of samples 1 and 6, which is $\sqrt{65}$. This is not less than $MinDist$, so we return to step 13ab) for the next test sample.

With test sample 2, the nearest sample $CNS(2)=3$ which is in the same cluster and the method proceeds to step 13abaa). The next distance $NXD(2)$ is the difference between the absolute distances of samples 2 and 5, which is $\sqrt{45}-\sqrt{5}$. This is less than the minimum distance $MinDist$, so $MeasDist$ is calculated as the distance to sample 5, which is $\sqrt{50}$. This is less than $CND(2)$, so $CNS(2) = 5$ and $CND(2) = MeasDist$. However, $MeasDist$ is not less than $MinDist$, so the method proceeds to step 13abad, $NXS(2)$ becomes 6, and the method returns to step 13abab. $NXD(2)$ is calculated as $\sqrt{65}-\sqrt{5}$. This is less than $MinDist$, so $MeasDist$ is calculated as $\sqrt{50}$. This is equal to $CND(2)$ and not less than $MinDist$, so the method proceeds to step 13abad, $NXS(2)$ becomes 7, and the method returns to step 13abab. $NXD(2)$ is $\sqrt{50}-\sqrt{5}$. This is less than $MinDist$, so $MeasDist$ is calculated as $\sqrt{61}$. This is not less than $MinDist$ or $CND(2)$, so the method proceeds to step 13abad, $NXS(3)$ becomes 8, and the method returns to step 13abab. $NXD(2)$ is $\sqrt{89}-\sqrt{5}$. This is not less than $MinDist$, so the method returns to step 13ab) for the next test sample.

With test sample 3, $NXS(3)=5$ and $NXD(3)=\sqrt{45}-\sqrt{5}$. This is less than $MinDist$, so $MeasDist$ is calculated as $\sqrt{26}$. This is less than $CND(3)$ and $MinDist$, so $CNS(3)=5$, $MinDist=CND(3)=MeasDist=\sqrt{26}$. At step 13abad, $NXS(3)$ is updated to 6 and $NXD(3)=\sqrt{65}-\sqrt{5}$. This is not less than $MinDist$, so the method returns to step 13ab) for the next test sample.

With test sample 4, $CND(4)=\sqrt{20}$, which is less than $MinDist$. Hence, $MinDist$ becomes $CND(4)$, and step 13a) is complete as this is the last test sample in the cluster.

At step 13b), sample 4 is the head and sample 6 is the tail. The corresponding edge is shown as h in Figure 3. Cluster 5 is merged into cluster 1 and the following information is recorded:

source(5)=4; dest(5)=6; height(5)=4.47; join(5)=1

As there is only one cluster left, the clustering method halts. The output of the clustering method comprises the array (source(i), dest(i), height(i), join(i)) as follows:

Index i	source(i)	dest(i)	height(i)	join(i)
2	1	2	2.24	1
3	3	2	2	1
4	4	3	3.61	1
5	4	6	4.47	1
6	6	7	1	5
7	5	7	4.12	5
8	8	7	1	5
9	9	8	2.24	5

Technical Processes

The method described above can be applied to any process or application involving a hierarchical clustering algorithm, preferably a single-link algorithm. However, the above method requires considerably fewer operations and therefore can be executed significantly faster, for a given platform, than known single-link clustering methods. The method can be applied to physical data samples, that is samples of physical quantities. The output of the method therefore represents an underlying physical structure of the physical quantities.

Some of the known applications will be described below, and categorised into generic types of process. Figure 6 shows the general form of apparatus for carrying out these processes, comprising a data input I, a pre-processing stage PRE, a clustering processor CP, a post-processing stage POST and a data output O. These stages do not necessarily represent discrete physical components. The data input I may be a sensor or

sensor array, or in the case of non-physical data, a data input device or network of such devices. The input data may be stored on storage means prior to further processing. The pre-processing stage PRE may perform analog-to-digital conversion, if necessary, and may also restrict the dimensions of the data to those required for clustering. The clustering processor CP, which may be one or many physical processors, performs the clustering method and outputs the clustering data. The post-processing stage POST may partition the single hierarchical cluster into multiple clusters for subsequent processing, in accordance with the clustering structure, and may perform automatic classification of the clusters based on their clustering structure and/or the properties of the data samples within the cluster. The data output O may be a display, a printer, or data storage means, for example.

Embodiments of the present invention include a program which performs a method in accordance with the invention when executed by a suitably arranged processor, such as the clustering processor CP. The program may be stored on a carrier, such as a removable or fixed disc, tape or other storage means, or transmitted and received on a carrier such as an electromagnetic signal.

Compression

Once a set of data samples has been classified into a hierarchical tree of clusters, the data samples themselves can be replaced by a compressed data set which describes the form of the clusters. In the case of lossy compression, the compressed data set represents the general form of the clusters without specifying the individual data samples. For example, the tree is separated into multiple clusters each having merging heights less than a predetermined value, and the data samples within each cluster are represented in the compressed data set by the coordinates of the centroid of the cluster. In the case of lossless compression, the tree is separated into multiple clusters each having merging heights less than a predetermined value, and individual samples within the cluster are represented by differential vectors from the centroid of the cluster; the differential vectors will have a smaller range than the absolute coordinates of the samples, and may therefore be represented using fewer bits. This technique is applicable to any type of data, whether physical data such as image, audio, video or quantities not directly perceptible by humans,

or non-physical data such as economic data. The compressed data set may be stored on a storage medium, giving more efficient storage, or transmitted over a communications link or local bus, giving reduced bandwidth requirements.

Hence, apparatus may be provided for carrying out this process, in which the data output O is a data store or data channel.

Segmentation and Feature Extraction

Once a set of data samples has been classified into a hierarchical tree of clusters, the tree may be divided into separate clusters, for example by setting a merge height at which the tree is divided. Each cluster represents a different class of data and may be used for analysis by attributing a different property to each cluster. The membership of each data sample to its cluster may be indicated, for example by colour-coding the samples in a display according to their cluster.

In the field of remote sensing, a similar technique may be used to display different object or terrain types. The display may be interpreted by a user or processed to provide automatic identification, for example by comparison with the shape or spectral properties of known object or terrain types.

A similar technique may be applied in image segmentation, where an image is partitioned into areas of similar colour or shade, for the purpose for example of converting a colour image into a greyscale image or a bitmap image into a vector image. This application is also an example of compression, in that the greyscale or vector image requires fewer bits than the original image.

In some cases, there will be an overlap between the segments of a data sample set, and it is then desirable to estimate the proportion of each segment type present in the overlap area. For example, in an image there may be areas which represent a mixing of two main components of the object of the image, such as trees and grass in a remote sensing image. Pure areas which contain only one component are first identified by finding data samples which are tightly clustered, and the spectral properties of these pure components are determined. Edge detection is then performed to identify the boundaries between these pure areas. In these boundary areas, the proportion of each component is then determined by fitting a mixing model of the spectral properties of the pure areas to the spectral

properties of the boundary areas. An example of this technique is described in the paper 'A Spectral Unmixing Algorithm for Distributed Endmembers with Applications to BioMedical Imaging' as referenced above.

If the properties of a data sample in the boundary areas cannot be fitted to within a given tolerance to the mixing model, the data sample is flagged as an anomaly and may be highlighted on a display. Anomaly detection is useful in medical imaging for the detection of small abnormalities, such as tumours, and in remote sensing for the detection of unusual objects or features.

In some specific applications, the boundary between pure samples is determined by the spatial or temporal properties of the samples. However, the boundary may be defined purely with reference to the clustering properties of the samples in their dimension space, so that tight clusters within that space are considered to contain pure samples, and samples between those clusters in the dimension space are considered to be mixed samples.

Hence, in apparatus which carries out this process, the post-processing step POST may generate data flags or labels associated with individual clusters, and the data output O provides an indication of the data flags or labels, such as a false colour or outline display of the data samples.

Data Mining/Browsing

In this technique, the cluster structure is used to select for inspection a subset of a large collection of data. In one case, one initial data sample is found and the other members of a cluster to which the initial data sample belongs are selected for inspection. One application of this technique is in the field of document searching and retrieval, such as web searching.

In another case, clusters are selected which have the desired properties, such as tight clustering (e.g. large numbers of samples with low merge heights) and the members of the selected clusters are inspected. One application of this technique is in the field of data mining, in which tight clusters within a large database are selected and analysed so as to make inferences based on the members of the cluster. Alternatively, the desired property may be a very loose clustering, for example in the field of fraud detection where a data sample that is dissimilar to other samples may indicate fraudulent activity.

Where the data samples are non-metric (i.e. they do not take the form of an array of measurement values), a dissimilarity function must be chosen so as to represent numerically the difference between any two data samples. The dissimilarity function may be a function of the number of similar words appearing in two documents, for example.

Hence, in apparatus for carrying out this process, the data input I may be a database and the data output O may be an identification of the selected subset of data, for example on a terminal.

Network Design

The minimum spanning tree represents a network connecting each data sample to at least one other sample such that the total edge length or weight is minimized, and therefore represents an optimum solution to real-life problems in which nodes need to be connected together in a network with maximum efficiency. Such problems include circuit design, in which the distance between data samples represents the length of wiring needed to interconnect circuit nodes. Similarly, where the data samples represent communications nodes and the distance between them represents the inefficiency incurred by interconnecting them, the minimum spanning tree represents the most efficient way of interconnecting the nodes. The method for finding the minimum spanning tree according to the above method may be applied to any such real-life problems.

Hence, in the apparatus for carrying out this process, the data input I may provide a data file representing the properties of nodes to be connected, and the data output O may represent a design for interconnecting the nodes. The design may be a graphic representation or a series of instructions to carry out the design. The series of instructions may be carried out automatically so as to create interconnections according to the design.

Pattern Recognition

Pattern recognition involves the classification of a new data sample based on its similarity to a set of data samples which have already been classified.

Cluster Rank Function

In pattern recognition, it is useful to generate a rank function for a given cluster of data samples. The rank function is a function of the dimensions of the data samples and gives a rank value of a new data sample as a member of the given cluster. The rank value can be used to determine in which cluster a new data sample should be classified.

A method will now be described for generating the rank function for a given cluster, given the data recorded above which defines the minimum spanning tree. The data which defines the minimum spanning tree need not be obtained by the clustering method described above, but this is preferable in view of the speed advantages.

Preprocessing – Reorder Output in Merge Height Order

For ease of subsequent processing, the output array is reordered in height order. In the specific example, the reordering is as follows:

Index <i>i</i>	<i>source(i)</i>	<i>dest(i)</i>	<i>height(i)</i>	<i>join(i)</i>
2	6	7	1	5
3	8	7	1	5
4	3	2	2	1
5	1	2	2.24	1
6	9	8	2.24	5
7	4	3	3.61	1
8	5	7	4.12	5
9	4	6	4.47	1

The result may be represented as shown in Figure 7, in which the samples are reordered to avoid any of the merge lines crossing.

Rank Function Contours

For ease of explanation, the contours of the rank function will now be described with reference to Figure 8, although it is not necessary to calculate the shape of the contours in order to calculate the value of the rank function for a new data sample.

First, hyperspheres (in our example, circles) having a radius equal to the smallest $height(i)$ are drawn around each sample joined at the smallest height. In our example the

hyperspheres are drawn around each of samples 6, 7 and 8. The perimeter of the overlapping hyperspheres is assigned a probability of $(N-y(1)-1)/(N-1)$, where $y(1)$ is the number of edges formed at that height. In this case, the probability is $6/8$.

We then proceed to the next smallest merge height of 2, which joins samples 2 and 3. Around those samples, we draw hyperspheres of radius equal to the next smaller merging height ($=1$), to which the perimeter is also assigned a probability of $6/8$. Next, around all of the samples of the current merge height or below, we draw hyperspheres of the current merge height, and assign to their perimeter a probability of $(N-y(2)-1)/(N-1)$, where $y(2)$ is the number of edges formed at the current merge height or below. In this case, the probability is $5/8$.

We then proceed to the next smallest merge height of 2.24, which applies to samples 1, 2, 8 and 9. We draw around them hyperspheres of radius equal to the next smaller merge height ($=2$), to which a probability of $5/8$ is also assigned. Next, around all of the samples of the current merge height or below, we draw hyperspheres of the current merge height and assign to their perimeter a probability of $(N-y(3)-1)/(N-1)$, where $y(3)$ is the number of edges formed at the current merge height or below. In this case, the probability is $3/8$.

We then proceed to the next smallest merge height of 3.61, which applies to samples 3 and 4. We draw around them hyperspheres of radius equal to the next smaller merge height ($=2.24$), to which a probability of $3/8$ is also assigned. Next, around all of the samples of the current merge height or below, we draw hyperspheres of the current merge height and assign to their perimeter a probability of $(N-y(4)-1)/(N-1)$, where $y(4)$ is the number of edges formed at the current merge height or below. In this case, the probability is $2/8$.

We then proceed to the next smallest merge height of 4.12, which applies to samples 5 and 7. We draw around them hyperspheres of radius equal to the next smaller merge height ($=3.61$), to which a probability of $2/8$ is also assigned. Next, around all of the samples of the current merge height or below, we draw hyperspheres of the current merge height and assign to their perimeter a probability of $(N-y(5)-1)/(N-1)$, where $y(5)$ is the number of edges formed at the current merge height or below. In this case, the probability is $1/8$. These circles are not shown in Figure 8, as there is insufficient space.

Finally, we reach the largest merge height of 4.47, which applies to samples 4 and 6. We draw around them hyperspheres of radius equal to the next smaller merge height (=4.12), to which a probability of 1/8 is also assigned. Next, around all of the samples of the current merge height or below, we draw hyperspheres of the current merge height and assign to their perimeter a probability of $(N-y(6)-1)/(N-1)$, where $y(6)$ is the number of edges formed at the current merge height or below. In this case, the probability is 0/8. These circles are not shown in Figure 8, as there is insufficient space.

To calculate the value of the rank function for a test data sample, we interpolate between the perimeters of the circles. Within the circles of smallest radius, we interpolate up to rank function=1 at the centre if the centres are the head and tail of the smallest edge. Otherwise, for the samples of next smallest merge height, the rank function is constant within circles of the smallest radius and set to the value at the boundary.

Rank Estimation – Spherical Case

The calculation of the rank value by interpolation will now be described in detail below, in a first order case as described above in which hyperspherical boundaries are defined.

Step 15 – Find Absolute Distance

For each sample, the radial or linear 'absolute distance' from the origin is calculated, as in step 2 above.

Step 16 – Sort Absolute Distances

The set of absolute distances of the samples is sorted and indexed.

Step 17 – Calculate Rank Values

The data samples are classified into one or more clusters. For example, the binary tree may be 'cut' at a specified merging height so that all clusters merged above that merging height are considered to belong to different clusters. Alternatively, the data samples may have been separated *a priori* into groups and clustering performed independently on each group.

For each cluster, as shown in outline in Figure 11:

17a) Find the sample NSP in that cluster nearest to the test sample, using a method similar to step 13a) for finding the nearest neighbour of a sample, but restricting the method to the current cluster and the test sample.

17b) Find the distance d from the test sample to the nearest sample NSP in the cluster, and the largest edge length e within the cluster. If the cluster contains only one sample and therefore no edges, make $e = d/2$. If $d > e$, the test sample is determined to lie outside the cluster altogether and the rank value $R = e - d$, which will be negative; the method then stops.

17c) The test sample lies within the cluster. If the cluster has only a single sample, make $R = 1 - d/e$ and stop.

17d) The cluster has multiple samples. Do the following steps:

17da) Let T be the number of edges in the minimum spanning tree of the cluster.

17db) Let CE be the index of the current edge under consideration; the edges are indexed in increasing length. CE is initially set to the first edge of height greater than or equal to the greater of d and $(r-d)$, where r is the merge height of the NSP.

17dc) Let NLE be the index of the first edge longer than the edge CE .

17dd) Let NH be the number of edges of length less than the edge CE .

17de) Let MIN be the distance from the test sample to the nearest sample considered so far; MIN is set to a maximum value initially.

17df) Let SL be the length of the longest edge in the MST shorter than the edge CE ; SL is initially set to zero.

17dg) Let LL be the length of the current edge CE ; LL is initially set to zero.

17dh) Let ND be the number of edges in the minimum spanning tree of the cluster with length equal to LL ; ND is set initially to 1.

17di) Find LL for the current edge CE .

17dj) Find ND . Increment CE by ND so that CE is now the index of the shortest edge of length greater than LL .

17dk) Define the set of 'active samples' AS as containing initially all samples with merge height less than or equal to SL. The 'active region' AR is defined as the region of all samples within a distance SL of any of the active samples AS.

17dl) If LL is less than or equal to the greatest edge length in the minimum spanning tree, find the updated AR and check whether the test sample falls within it as follows:

17dla) Add the samples with merge height LL to the active samples AS. For each of the samples with merge height LL, measure the distance TD to the test sample. If TD is less than MIN, set MIN as the greater of TD and SL.

17dlb) If MIN is less than or equal to LL, go to step 17dn) to find the rank value, as the test sample lies within the active region.

17dlc) Set SL=LL and let LL be the length of the new edge. Add ND to NH, as the number of edges of length less than LL has increased by ND. Set ND to be the number of edges in the minimum spanning tree of length LL. Add ND to CE so that CE is the index of the shortest edge of length greater than LL.

17dm) Go back to step 17dl).

17dn) The rank value R is given as follows:

$$R = \frac{LL - MIN}{LL - SL} \times \frac{ND}{T} + \frac{T - ND - NH}{T} \quad (1)$$

Rank Estimation – Ellipsoidal Case

The calculation of the rank value by interpolation will now be described in an alternative second-order case in which hyperellipsoidal boundaries are defined. Instead of defining spheroidal boundaries from each sample as shown in Figure 8, ellipsoidal boundaries are defined with the samples at either end of an edge as the foci. Step 17 is replaced by step 17' as follows:

17a') Find the sample NSP in that cluster nearest to the test sample, using a method similar to step 13a) for finding the nearest neighbour of a sample, but restricting the method to the current cluster and the test sample.

17b') Find the distance d from the test sample to the nearest sample NSP in the cluster, and the largest edge length e within the cluster. If the cluster contains only one

sample and therefore no edges, make $e = d/2$. If $d > 1.5 \times e$, the test sample is determined to lie outside the cluster altogether and the rank value is assigned a value as follows:

17ba') Using the indexed absolute distances to order and limit the search, find the edge in the minimum spanning tree of the cluster which has the smallest string distance S to the test sample. The two samples connected by the edge can be found from the source(i) and dest(i) arrays. If the smallest current S is r , then any edge with a smaller S must have at least one of its samples lying within an absolute distance of $1.5 * r$ of the test sample; hence, the search is limited to this range. d takes the value of the smallest S , and $R = 1 - e/d$. The procedure then stops.

17c') The test sample lies within the cluster. If the cluster has only a single sample, make $R = 1 - d/e$ and stop.

17d') The cluster has multiple samples. Do the following steps:

17da') Let T be the number of edges in the minimum spanning tree of the cluster.

17db') Let CE be the index of the current edge under consideration; the edges are indexed in increasing length. CE is initially set to the first edge of height greater than or equal to two thirds of the greater of d and $(r-d)$, where r is the merge height of the NSP.

17dc') Let NLE be the index of the first edge longer than the edge CE .

17dd') Let NH be the number of edges of length less than the edge CE .

17de') Let MIN be the distance from the test sample to the nearest sample considered so far; MIN is set to a maximum value initially.

17df') Let SL be the length of the longest edge in the MST shorter than the edge CE .

17dg') Let LL be the length of the current edge CE ; LL is initially set to zero.

17dh') Let ND be the number of edges in the minimum spanning tree of the cluster with length equal to LL ; ND is set initially to 1.

17di') Find LL for the current edge CE .

17dj') Find ND. Increment CE by ND so that CE is now the index of the shortest edge of length greater than LL.

17dk') Define the set of 'active edges' AE as having no samples initially. The 'active region' AR is defined as the region of all samples within a 'string distance' S of LL of any edge of AE. The 'string distance' is one half of the difference between the sum SD of the distances from the test sample to each of the samples connected by the edge and the length of the edge:

$$S = (SD - LL) / 2$$

17dl') Find the updated AR and check whether the test sample falls within it as follows:

17dla') Add the samples with merge height LL to the active samples AS. For each of the samples with merge height LL, measure the string distance S to the test sample. If S is less than MIN, set MIN as the greater of S and SL.

17dlb') If MIN is less than or equal to LL, go to step 17dn') to find the rank value, as the test sample lies within the active region.

17dle') Set SL=LL and let LL be the length of the new edge. Add ND to NH, as the number of edges of length less than LL has increased by ND. Set ND to be the number of edges in the minimum spanning tree of length LL. Add ND to CE so that CE is the index of the shortest edge of length greater than LL. If there are no longer edges in the minimum spanning tree, then the test sample lies outside the cluster; so go to 17dn').

17dm') Go to 17dl')

17dn') The estimated rank value is given by:

$$R = \frac{LL - MIN}{LL - SL} \times \frac{ND}{T} + \frac{T - ND - NH}{T} \quad (1')$$

17do') The negative estimated rank is given by $R = MIN - SL$

Classification

Where there is more than one cluster, the test sample is assigned to the cluster having the greatest estimated rank value.

This process has many practical applications in the general field of artificial intelligence involving automatic recognition of an input, such as an image or sound. For example, in a voice recognition application a series of training samples are input at the data input I during a training phase. It is known *a priori* what sounds or phonemes the training samples are intended to represent, so the samples are divided according to intended classification and each classification of samples is clustered independently. In recognition mode, a rank value is calculated for a test sample in relation to each of the clusters and the test sample is classified according to comparison between these rank values. A 'hard' classification may be made by assigning only one classification to the test sample, or a 'soft' classification may be made by assigning a probability of the test sample belonging to each of a number of different classifications. The 'soft' classifications may be used to classify a series of sounds by determining the relative probabilities of possible sequences of sounds, weighted by the 'soft' classification probability of each sound.

This technique has the advantage that the envelopes of the clusters may partially overlap that of another cluster in dimension space, while still allowing a classification decision to be made.

The technique may be applied to data samples each representing a spatial configuration, for example in the field of optical character recognition (OCR), in which the output is a representation of a recognised character, or robotic vision in which the output is one or more actions performed in response to the classification of the test sample.

Hence, in apparatus for carrying out this process, the data input I is a sensor or sensor array, and the data output O is a representation of the classification of an input, which affects subsequent processing steps by the apparatus.

Specific Example

A specific example of estimating the rank value of a test sample will now be described with reference to Figure 9. The single cluster of Figure 8 is divided into two clusters by removing the longest edge between samples 4 and 6; this can be represented as cutting the binary tree at a height between 4.12 and 4.47. A test sample TP is given at coordinates (6,3) and it is desired to find the rank value of the test sample for each cluster, for the purpose of determining to which of the two clusters the test sample should belong.

The rank contours are now as shown in Figure 9. By way of comparison, the rank contours for the ellipsoidal case are as shown in Figure 10.

Following the hyperspheroidal case as described above, we take each cluster in turn. In the cluster of samples (1, 2, 3, 4), the closest sample to the test sample is sample 4, for which d is $\sqrt{5}$. e is $\sqrt{10}$, so TP lies within this cluster. There are 3 edges, in increasing length: b, a, c. CE initially points to edge a, since this has a length equal to d , and $LL = \sqrt{5}$. Samples 1, 2 and 3 are added to the active samples. Sample 3 is closer to TP, so MIN becomes $\sqrt{17}$. This is greater than LL , so we find the next longer edge, which is edge a, of length $\sqrt{5}$. Now we return to step 17d1) and add sample 1 to the active samples. However, sample 1 is further from TP than the samples already tested, so MIN is still $\sqrt{17}$, which is greater than LL . We therefore find the next longer edge, which is edge c. Sample 4 is added to the active samples, and $LL = \sqrt{10}$, while $SL = \sqrt{5}$. MIN becomes $\sqrt{5}$, which is less than LL ; hence, we calculate:

$$R = \frac{\sqrt{10} - \sqrt{5}}{\sqrt{10} - \sqrt{5}} \times \frac{1}{3} + \frac{3 - 1 - 2}{3} = \frac{1}{3} \quad (2)$$

as can be confirmed by observing that TP lies on the $R=1/3$ boundary.

Although the above embodiments have been described with reference to a Euclidean metric, it will be appreciated that other types of metric may alternatively be used. Moreover, aspects of the present invention may be applied to non-metric data samples, and to clustering methods other than single-link clustering.

4 Brief Description of Drawings

Specific embodiments of the present invention will now be described with reference to the accompanying drawings, in which:

Figure 1 is a flowchart showing the principal steps of a method in an embodiment of the present invention;

Figure 2 is a flowchart showing the detailed steps of step 13a of the flowchart of Figure 1;

Figure 3 shows the minimum spanning tree of a set of data samples, calculated using the method shown in Figures 1 and 2;

Figure 4 shows a binary tree at an intermediate stage of the calculation of the minimum spanning tree;

Figure 5 shows the binary tree at the final stage of the calculation;

Figure 6 is a generic diagram of apparatus for carrying out the method in a technical process;

Figure 7 shows the binary tree of Figure 5 sorted in order of merge height;

Figure 8 shows the hyperspherical contours of a rank value function for a cluster of the data samples of Figure 3;

Figure 9 shows the hyperspherical contours of a rank value function for a sub-cluster of the data samples of Figure 3;

Figure 10 shows the hyperellipsoidal contours of a rank value function for the sub-cluster of Figure 9; and

Figure 11 is a flow diagram of the calculation of the rank value function.

Fig. 1

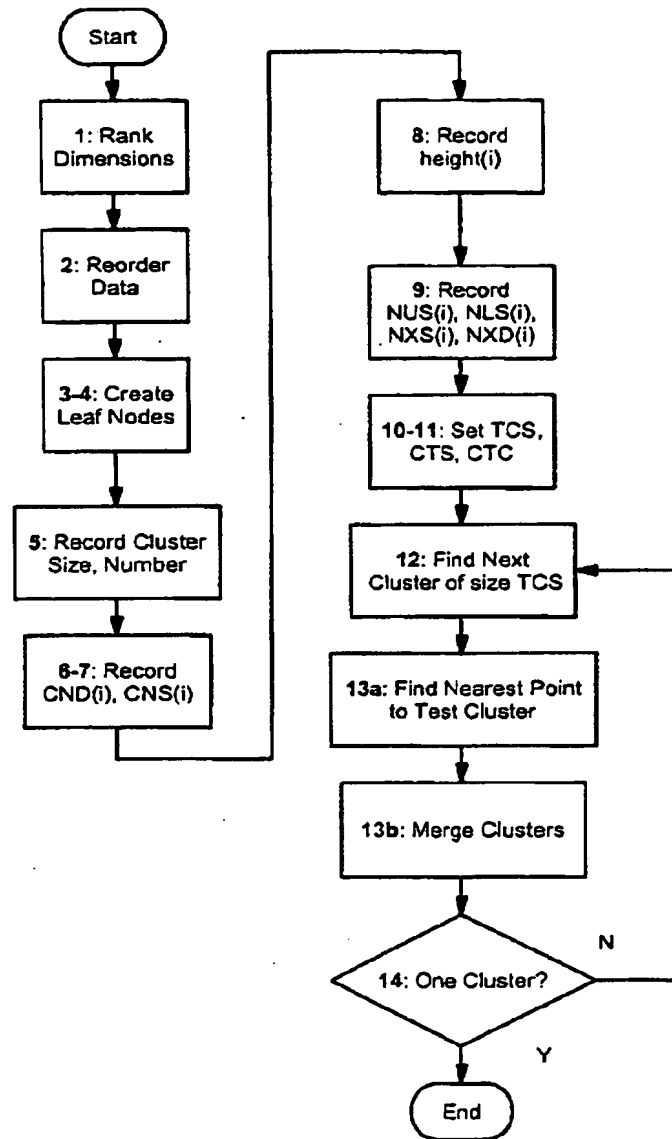
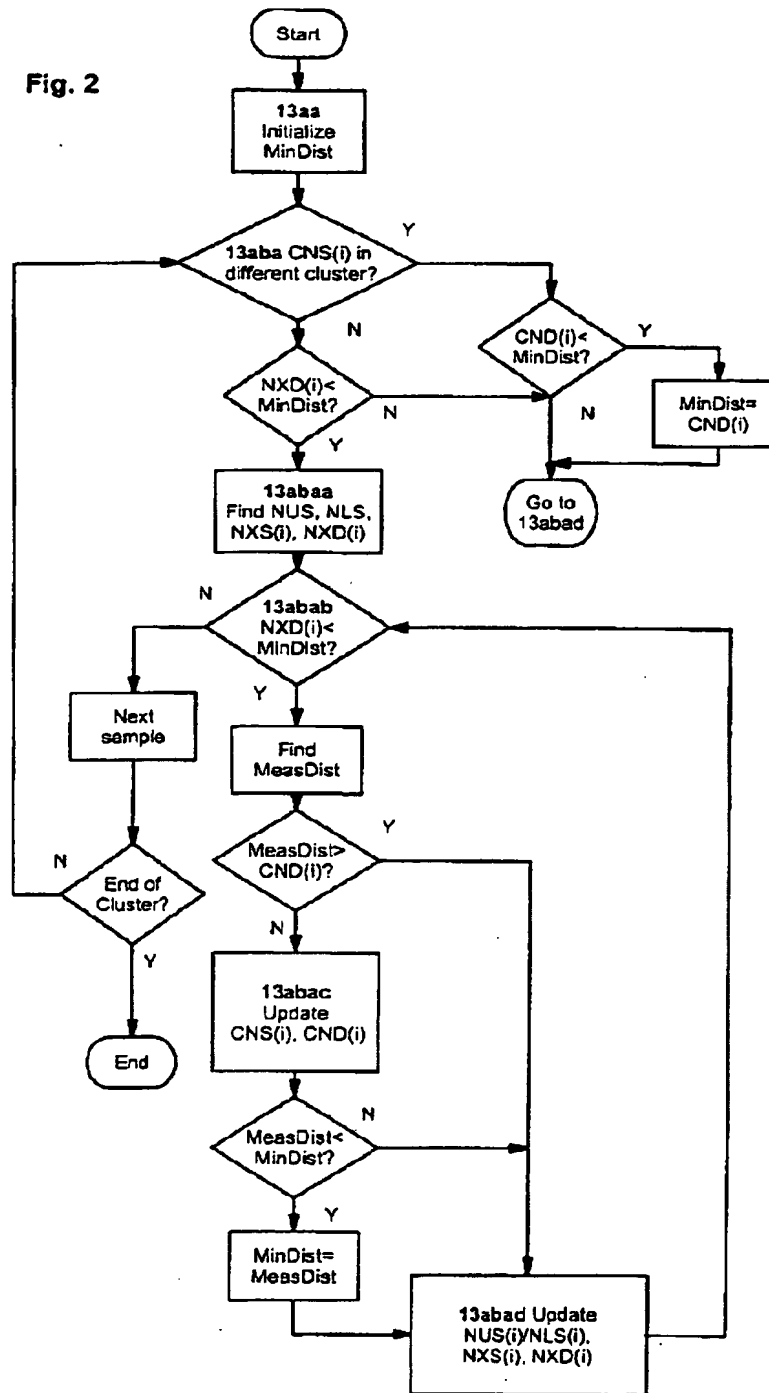


Fig. 2



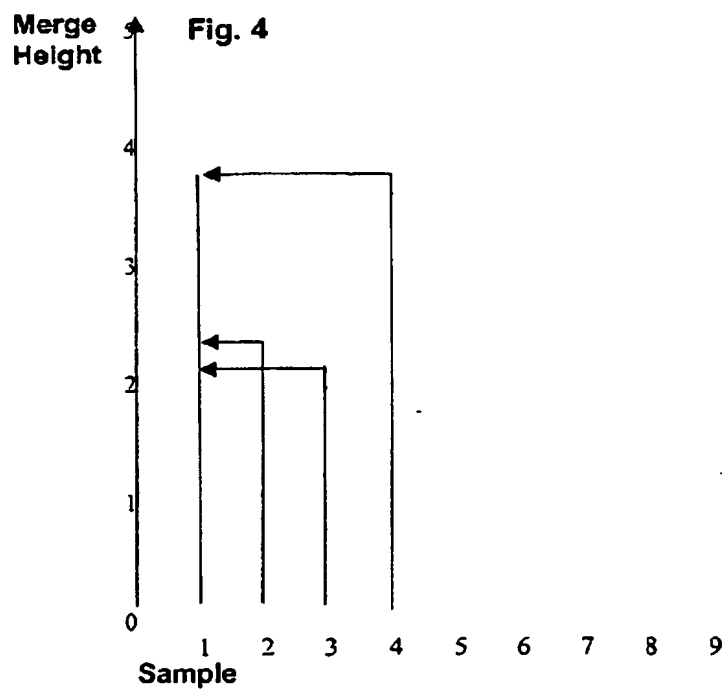
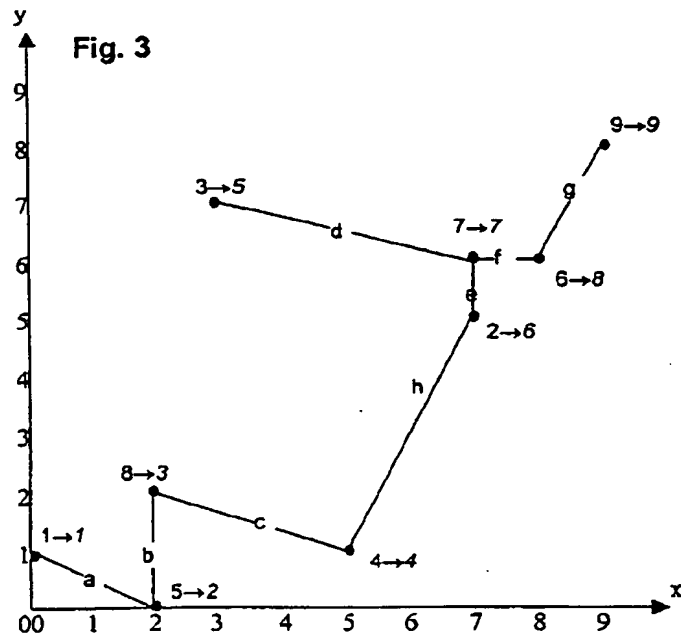


Fig. 5

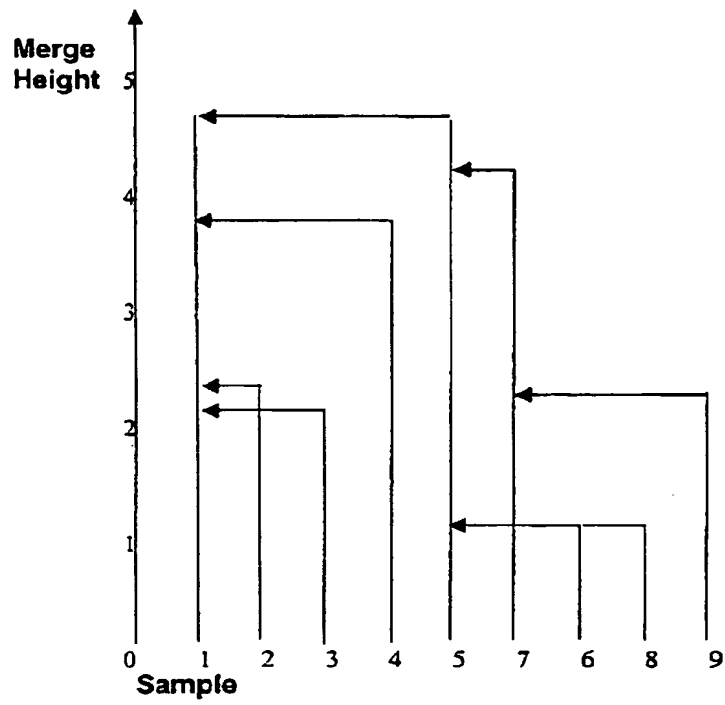
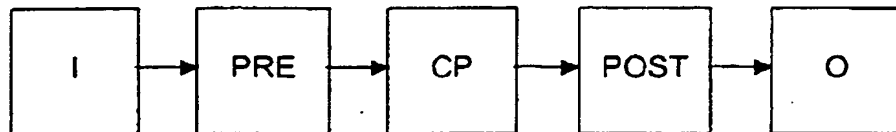


Fig. 6



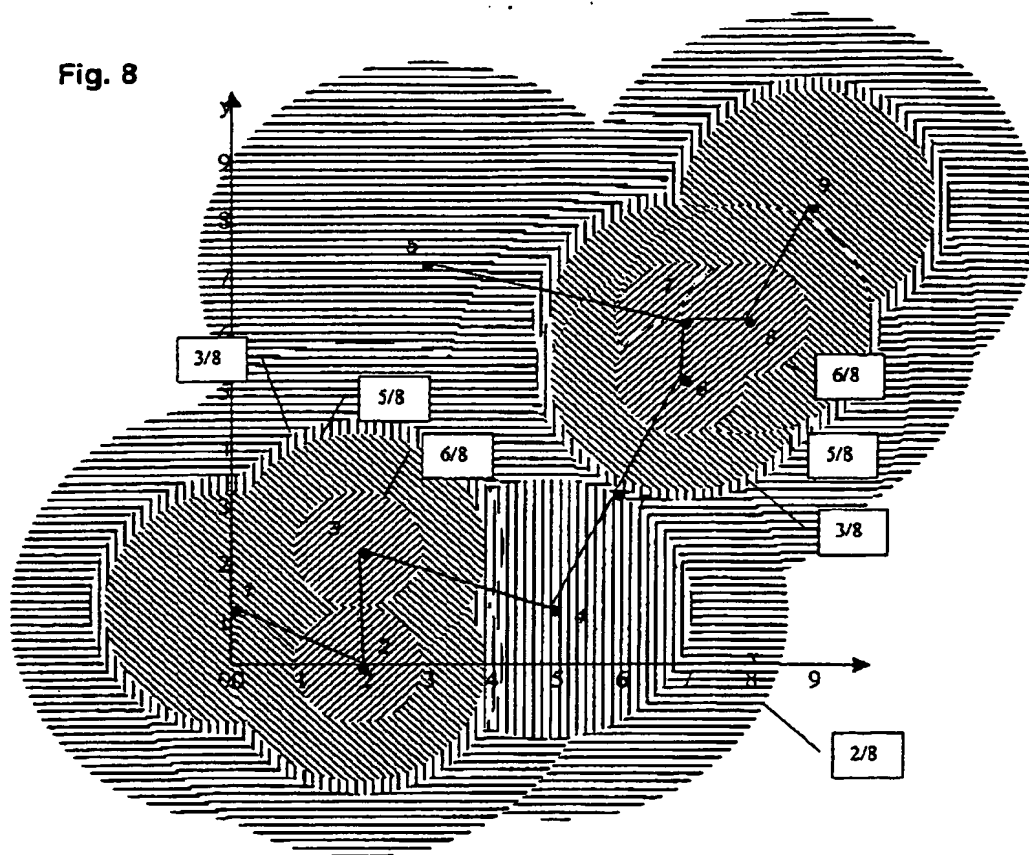
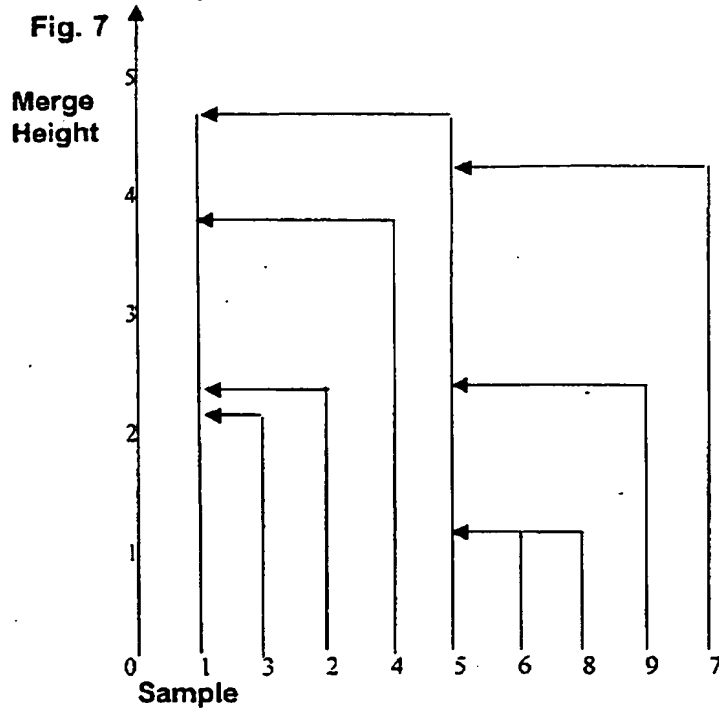


Fig. 9

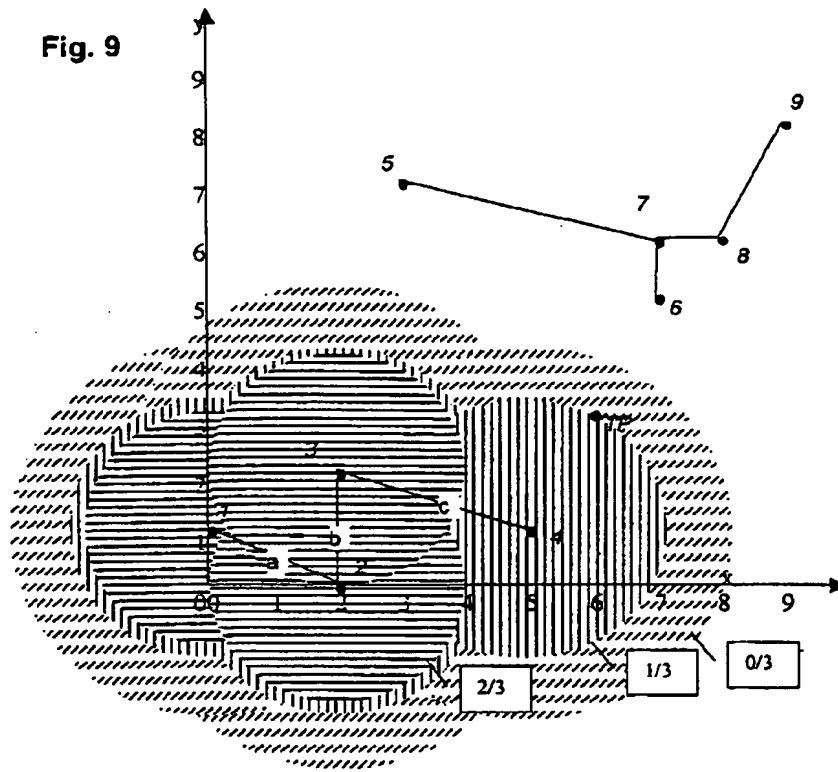


Fig. 10

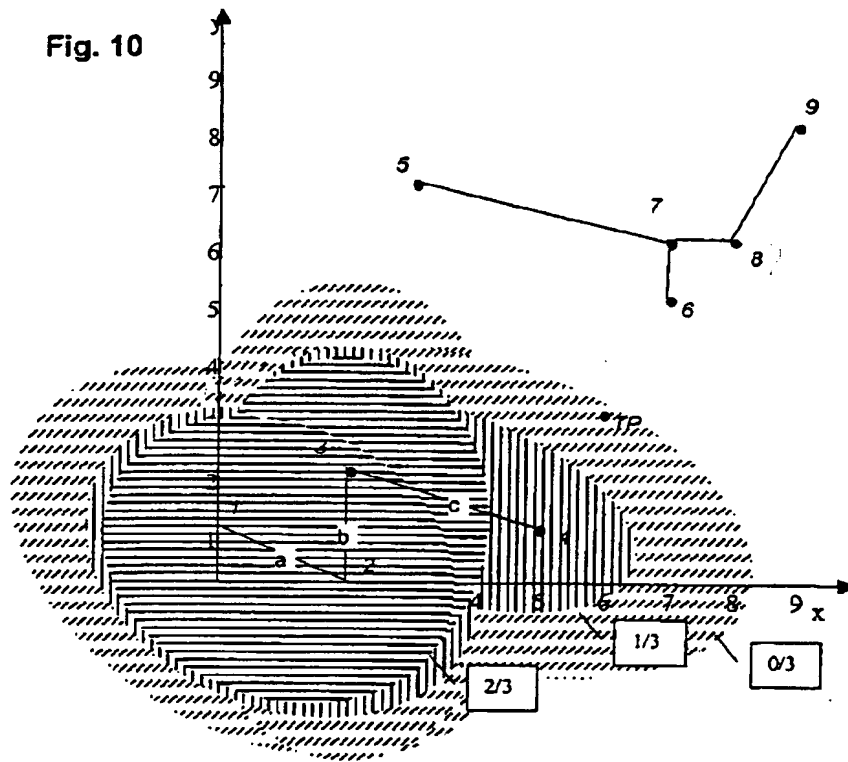
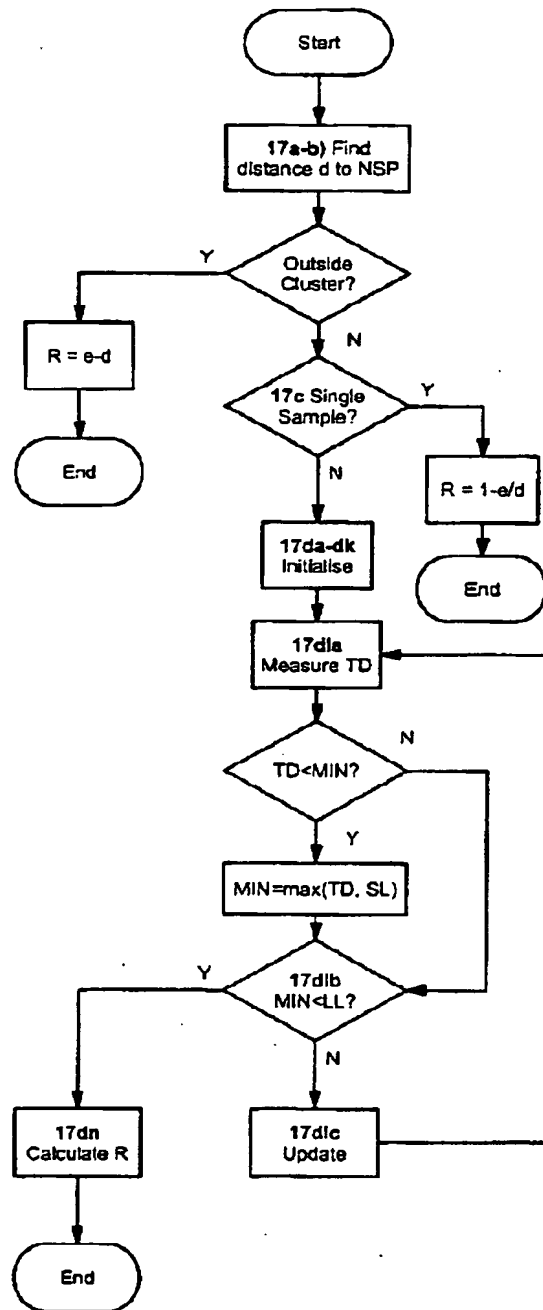


Fig. 11



A data clustering method involves techniques for improving the speed of generation of clustering data representing hierarchical clustering of a set of data samples. The techniques include the selection of clusters in increasing size for selecting the nearest other cluster for merging, ordering the data samples according to absolute distance from a reference and searching for nearest neighbours within a restricted index range, and making distance comparisons by summing the contributions from components in each dimension in turn in order of the interquartile ranges of components of the data samples in each dimension. A data classification method involves calculating a rank value for a test sample in relation to a cluster of data samples, by taking into account the dissimilarities of the data samples at either end of the closest edge to the data sample and/or by calculating as a function of a test sample dissimilarity of the test sample to the most similar data sample within the cluster, unless the test sample dissimilarity is less than the dissimilarity of an edge in a minimum spanning tree which has the greatest dissimilarity less than an edge connected to the most similar data sample.

The applications of the methods include data compression, feature extraction, unmixing, data mining and browsing, network design and pattern recognition.

2 Representative Drawing Fig. 1

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☒ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.